Fault detection in a wastewater treatment plant based on neural networks and PCA

M.J. Fuente, D. Garcia-Alvarez, G.I. Sainz-Palmero, P. Vega

Abstract— **In this paper, a neural network PCA method that integrates neural networks (NN) and principal component analysis (PCA) is used to detect faults in a wastewater treatment plant. The neural networks are used to calculate a non-linear and dynamic model of the process in normal operating conditions. PCA is used to generate monitoring charts based on the residuals calculated as the difference between the process measurements and the output of the networks. It can evaluate the current performance of the process and detects the faults. This technique has been applied to the simulation of a benchmark of a biological wastewater treatment process, a highly non-linear process. The simulation results clearly show the advantages of using this NNPCA monitoring in comparison with classical PCA monitoring.**

Index Terms— **Fault detection, Process monitoring, Principal Component Analysis, Neural networks, Wastewater treatment plants.**

I. Introduction

On-line monitoring and diagnosis of chemical processes are essentially required for plant safety, product quality and maintenance of the process equipment. Modern control techniques have solved a large number of problems, but when a special cause occurs in a process, it cannot operate under control. The development of an industrially reliable online scheme for such processes would be a step toward effectiveness and robustness. For most chemical processes, modern computers provide a system in which hundreds of variables may be recorded and stored cheaply and efficiently. These data can be analyzed to determine whether or not a fault has occurred in the process, associated with equipment failure, equipment wear, extreme process disturbances, etc. Although the process data are accessible from any time period, it is difficult for everyone to find out the process problems only by examining the data because of the large amount of data, the existence of multiple variables and the high correlation between the variables. So, this implies a strong need to develop tools that help the operators to detect and diagnose the problems in the process.

Several techniques have been developed for monitoring, fault detection and diagnosis. These techniques can be classified into three categories: quantitative model-based methods, qualitative models and search strategies, and process history-based methods ([1], [2], [3]). The first category is also called analytical redundancy (AR) ([4], [5]), in which an explicit input-output model is used. Unusual events are detected by referencing the measured process behavior against the model. The AR approach works well when an explicit model of the process is available. However, such a model is not easily obtained due to nonlinearity, complexity and the large size of the processes. The third category is the history-based methods, in which multivariate statistical techniques, such as principal component analysis (PCA) or partial least squares (PLS), are used to monitor the process ([6], [7]). In this case, an implicit statistical model is built using data obtained when the process is operating well and under control. Unfortunately, these two methods are only good for linear or closed-linear processes and they fail in non-linear and dynamic systems. The monitoring techniques based on neural networks also fall into this category. In the task of fault detection and diagnosis, the neural networks can be used as a classifier of faults based on process measurements or as an alternative to the traditional model estimator, i.e., to obtain a non-linear model of the process ([8]).

Most multivariate statistical monitoring methods based on PCA implicitly assume that the observations at one time instant are statistically independent of observations at a past time. However, this situation does not occur in industrial plants. These effects give the variables' autocorrelation and the system dynamic properties. In order to solve this problem, i.e., to take into account the serial correlation in the data or to capture the process dynamic, [9] proposes dynamic PCA (DPCA), which uses an augmented matrix with time-lagged variables, while [10] identifies the system states and the state space model parameters using the multivariate statistical projection techniques of canonical variate analysis (CVA) and PLS. To solve the problem of non-linearity, some extensions of non-linear PCA have been reported in the literature, as [11], which develops a nonlinear PCA based on autoassociative neural networks, and [12] proposes a nonlinear PCA by combining the principal curves and neural networks. Also, [13] uses, the just-in-time learning method (JITL) and principal component analysis to construct a JITL-PCA monitoring scheme, where the JITL method is used to calculate a non-linear and dynamic model of the process in normal operating conditions, while the residuals are also analyzed by PCA to evaluate the status of the current process operation.

In this paper, a general monitoring method that integrates neural networks and PCA is applied to a wastewater treatment plant (WWTP). The neural networks are used to model the nonlinear dynamic system in nominal operating condi-

M.J. Fuente, D. Garcia-Alvarez and G.I. Sainz-Palmero are with the Department of Systems Engineering and Control. University of Valladolid, EII, C/ Paseo del Cauce s/n 47011 Valladolid, Spain `[maria@autom, dieggar@cta gresai@eis].uva.es`

P. Vega is with the Department of Computing and Automation, University of Salamanca, ETSII, Av. Fernando Ballesteros 2, 37700, Bejar, Salamanca, Spain `pvega@usal.es`

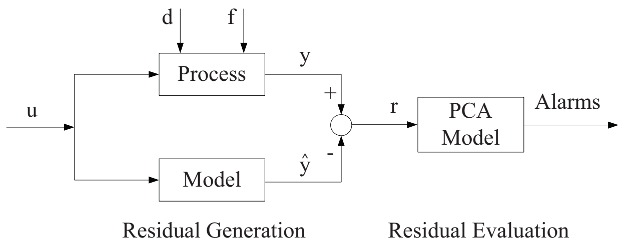


Fig. 1. Model-based residual generator and PCA residual evaluation

tions. The actual behavior of the system is compared with the output of the neural network through the same observations, calculating the residuals. Then, these residuals are evaluated by the PCA technique, using the classical charts, T^2 and Q, to detect the faults, and the contribution analysis [14] to perform the fault isolation. The neural networks are used to remove the non-linearities and the dynamic characteristics of the process.

The organization of the paper is as follows. First, a description of the dynamic process monitoring integrating neural networks and PCA is explained in section 2. The superiority of this monitoring method in detecting and identifying faults over PCA is illustrated in section 3 through the application to a wastewater treatment simulation benchmark. Finally, the conclusions are given in section 4.

II. NNPCA MONITORING SCHEME

The process monitoring structure and the classical fault detection and diagnosis schemes consist essentially of two stages: residual generation and residual evaluation. The residuals are generated as the difference between the current behavior of the process and the output of a nominal model driven by the same inputs. It is expected that these residuals are close to zero in nominal conditions and drift away when there exists a fault in the process. There are many ways to calculate the model for the residual generator [5]. In this paper the mathematical model used to generate the residuals is a neural network, and the residual evaluation is carried out with the PCA monitoring technique instead of simply comparing the residuals with a threshold (Fig. 1).

A. Dynamic neural networks for residual generation

For complex real processes, it is very hard to implement a first principle model that summarizes the relationship between the variables. In these cases, an empirical model can approximate the process behavior from time series data without explicit knowledge of the process. So, many methods have been proposed to model non-linear processes driven by data [15], one of them being the neural networks approach.

In general, the neural networks with multiple inputs and single output (MISO) can generally be used to model any non-linear system. Dynamic characteristics of complex systems can often be inferred from the analysis of time series calculating different types of models such as NARX (Nonlinear ARX) or NOE (Nonlinear OE). In this case, the model used for the output $y_j(k)$ is the NOE and can be expressed as:

$$\begin{aligned}\widehat{y}_j(k) \;=\; & NN(\widehat{y}_j(k-1),\widehat{y}_j(k-2),\ldots,\widehat{y}_j(k-n), \\ & u_1(k-1),u_1(k-2)\ldots,u_1(k-m),\ldots, \\ & u_p(k-1),u_p(k-2),\ldots,u_p(k-m))\end{aligned} \quad (1)$$

where $\widehat{y}_j(k)$ and $u_i(k)$ are the j-th predicted output for the model and the i-th input respectively, n indicates the number of delays in the predicted output j used as inputs to the network, m indicates the number of delays in all of the inputs and p is the number of inputs considered. NN is the non-linear neural network used that, in this case, has a feedback connection from the predicted output to the inputs. The NN will extract true dynamics from the noise data, and ideally the prediction error will be only the measurement noise. This NN is trained with data under normal operating conditions. When the process output is compared with the outputs of the neural networks, the residual can be calculated as:

$$\begin{aligned} r(k) &= [r_1(k)\ r_2(k)\ \ldots\ r_n(k)]^T = \\ &= [y_1(k)-\widehat{y}_1(k)\ y_2(k)-\widehat{y}_2(k)\ \ldots\ y_n(k)-\widehat{y}_n(k)]^T \end{aligned} \quad (2)$$

Then, the evaluation of the residuals is carried out by the PCA technique.

B. PCA for residual evaluation

Principal Component Analysis (PCA) is one of the most widely used multivariate statistical techniques. PCA is based on a linear transformation that produces new uncorrelated variables (components) from the original correlated measured variables. PCA provides a method of extracting relevant information from huge noisy data sets. This extraction implies a dimensionality reduction of the original data, so that a few of these components are sufficient to adequately represent the hidden sources of variability in the process.

Mathematically, PCA estimates the correlation structure of the process variables. The size of the variance residual indicates the importance of every variable. Consider a data matrix $\mathbf{X} \in \Re^{K \times J}$ containing K samples of J process variables collected under normal operation and scaled to zero mean and unit variance. The covariance matrix of $\mathbf{X}$ is defined as:

$$\mathbf{S} = \frac{1}{K-1}\mathbf{X}^T\mathbf{X} \quad (3)$$

and performing the singular value decomposition SVD [16]:

$$\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^T \quad (4)$$

the transformation matrix $\mathbf{P}_{1:A} \in \Re^{J \times A}$ is generated by choosing A eigenvectors or columns of V corresponding to A first principal eigenvalues. Now the prediction for the PCA model for a new observation data $\mathbf{x}_{new}$ is given by:

$$\hat{\mathbf{x}}_{new} = \mathbf{P}_{1:A}\mathbf{t} \quad (5)$$

where $\mathbf{t} = \mathbf{P}_{1:A}^T \mathbf{x}_{new}$ is the score vector. The resulting residual is defined as:

$$\mathbf{r}_{PCA} = \mathbf{x}_{new} - \widehat{\mathbf{x}}_{new} = (\mathbf{I} - \mathbf{P}_{1:A}\mathbf{P}_{1:A}^T)\mathbf{x}_{new} \tag{6}$$

When PCA is applied to fault detection tasks, two monitoring statistics can be used. These statistics are used in control charts. These charts can detect a fault when their values are greater than a determinate threshold. Hotelling's T^2 and square prediction error (SPE or Q) are the most popular. The monitoring can be reduced to these two variables (T^2 and Q). T^2 can be calculated for a new measured variables vector $\mathbf{x}$ as follows:

$$T^2 = \mathbf{x}^T \mathbf{P}_{1:A} \Lambda_A^{-1} \mathbf{P}_{1:A}^T \mathbf{x} \tag{7}$$

where Λ_A is a squared matrix formed by the first A rows and columns of Λ.

The process is considered *normal* for a given significance level α if:

$$T^2 \leq T_\alpha^2 = \frac{(K^2-1)A}{K(K-A)} F_\alpha(A, A-K) \tag{8}$$

where $F_\alpha(A, K-A)$ is the critical value of the Fisher-Snedecor distribution with A and $K-A$ degrees of freedom and α is the level of significance. α takes values between 90% and 95% [17].

The Q statistic is calculated as the sum of the squares of the residuals:

$$Q = \mathbf{r}_{PCA}^T \mathbf{r}_{PCA} \tag{9}$$

The upper limit of this statistic can be computed as follows:

$$Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{10}$$

with:

$$\theta_i = \sum_{j=R+1}^{J} \lambda_j^i \qquad h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$$

where c_α is the value of the normal distribution, with α being the level of significance [17].

In this paper, instead of the original observation variables, the PCA technique described here is applied to the residuals calculated in eq. (2), i.e., the residual data at normal operation is computed from the predicted neural network and arranged in the matrix $\mathbf{R} \in \Re^{K \times N}$ with K samples and N prediction residuals variables, and then the PCA technique is applied to this new matrix $\mathbf{R}$ instead of $\mathbf{X}$.

III. Application to a Wastewater Treatment Plant

The approach presented in this paper has been tested in a simulated wastewater treatment plant (WWTP). This plant is based on the COST benchmark [18]. This benchmark was developed for the evaluation and comparison of different activated sludge wastewater treatment control strategies. The model is implemented using MATLAB© and SIMULINK©.

Fig. 2 shows an overview of this plant. It is composed of a two-compartment activated sludge reactor consisting of two anoxic tanks followed by three aerated tanks. This type of plant combines nitrification with predenitrification in a configuration that is usually built for achieving biological nitrogen removal in full-scale plants. The reactor is followed by a secondary settler. The settler is modeled as a 10 layer non-reactive unit. The mathematical model of the plant and the parameters can be found in [18].

The influent used was the dry and rainy influent data files [18]. In these files, the variation of influent flow is between $15000 - 35000\, m^3/d$. The plant, as Fig. 2 shows, has two refluxes: external reflux, from settler to input, which is approximately equal to the influent flow, and internal reflux, from the last aerated tank to input, which is approximately equal to three times the influent flow, but this is a manipulate variable. The objective of the control strategy is to control the dissolved oxygen level in the aerated reactor by manipulating the oxygen transfer coefficient (K_La_5) and to control the nitrate level in the anoxic tank by manipulating the internal recycle flow rate. The controllers are of PI type.

The variables involved in this work are concentrations of:

1) Alkalinity (S_{ALK}).
2) Soluble biodegradable organic nitrogen (S_{ND}).
3) Ammonia nitrogen (S_{NH}).
4) Nitrate (S_{NO}).
5) Dissolved oxygen (S_O).
6) Readily biodegradable substrate (S_S).
7) Active autotrophic biomass ($X_{B,A}$).
8) Active heterotrophic biomass ($X_{B,H}$).
9) Particulate biodegradable organic nitrogen (X_{ND}).
10) Particulate products from biomass decay (X_P).
11) Slowly biodegradable substrate (X_S).
12) The effluent flow rate (Q_0).
13) The internal recycle flow rate (Q_{int}).
14) The oxygen transfer coefficient ($K_{La}5$).

In this case, three faults have been considered. They are not sensor or actuator faults, they are faults in the process, i.e., multiplicative faults. The faults considered are:

- **Toxicity shock (T)** This fault is due to a reduction in the normal growth of heterotrophic organisms. This type of fault can be produced by toxic substances in the water coming from textile industries or pesticides. This fault is simulated by reducing the maximum heterotrophic growth rate (μ_H).
- **Inhabitation (I).** This fault can be produced by hospital waste that can contain bactericides, or metallurgical waste that can contain cyanide. This type of fault is due

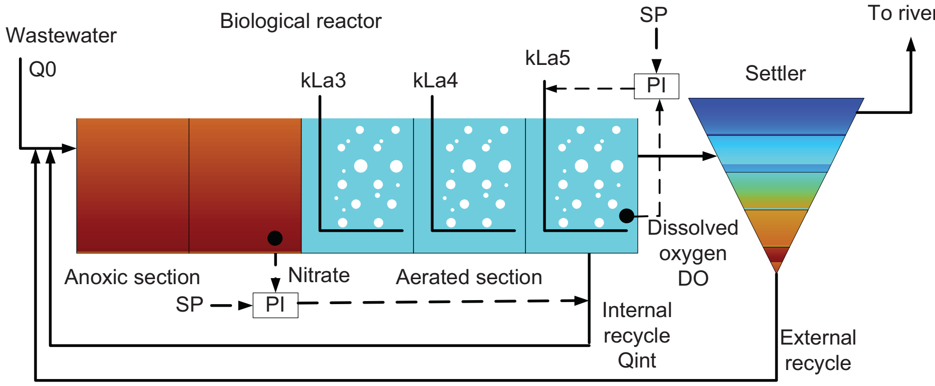


Fig. 2. General overview of the wastewater treatment plant (WWTP)

to a reduction in the normal growth of the heterotrophic organisms and an increase in the decay factor of this type of organisms. In this case, the fault is caused by reducing the maximum heterotrophic growth rate (μ_H) and increasing the heterotrophic decay rate (b_H).

- **Bulking (B).** This type of fault is produced by the growth of filamentous microorganisms in the active sludge. This phenomenon causes the impossibility of decantation in the settler. To simulate this fault the settling velocity (v_{sj}) is reduced.

The plant model was simulated for 100 days in open-loop configuration in order to determine the steady state. Then, the simulation in closed-loop was simulated for 28 days, 100 samples being taken per day. Two data sets in normal conditions were collected from the plant, one for training the neural networks and the other for testing the model calculated, modifying, in both data sets, the manipulated variables, i.e., the control inputs to the plant (Q_{int} and K_{La5}).

The neural network model relates the current measured variable to all the past measured and predicted variables. To determine the network's architecture, the measured variable lags and the number of hidden neurons are obtained by the appropriate adjustment based on the prediction error. Eight neural networks are constructed for the principal variables of the process: S_{ALK}, S_{ND}, S_{NH}, S_{NO}, S_O, S_S, $X_{B,A}$ and $X_{B,H}$, taking the rest of variables as inputs to the different networks, as explained before in Eq.1. Table I shows the order and structure of the eight neural networks considered. All neural networks have one hidden layer with n_{hd} neurons. Then, the multivariable residuals are obtained by computing the differences between the actual measurements of the process to be supervised and the prediction of the neural network model. After that, the PCA is applied to the residual data and the two control charts explained before are constructed to detect the actual behavior of the system. The PCA model was established with 5 components. Also, if a fault is detected, the contribution plots are applied to diagnose which variable is the root cause of the fault.

Also, for comparison purposes, a classical PCA monitoring scheme was constructed for this process, taking the 14 variables described above. This monitoring scheme was

TABLE I

ORDER AND STRUCTURE OF THE EIGHT NEURAL NETWORKS

Variable	n_{hd}	m	n
S_{ALK}	10	2	3
S_{ND}	10	1	2
S_{NH}	10	2	3
S_{NO}	5	1	2
S_O	7	1	2
S_S	7	2	3
$X_{B,A}$	10	2	3
$X_{B,H}$	5	2	3

performed using a classical PCA model with 4 components. The effectiveness of this monitoring scheme was tested using a set of faulty data different from the nominal data which had been used to train the networks and to calculate the classical PCA model. For each fault operation, the simulation started in nominal conditions and the faults were introduced in different sampling times.

The method has been tested with the three different faults described above with different fault sizes. All the obtained results can be seen in Table II, where the experiments with different faults, at time $t = 200$ and different sizes are summarized. The fault detection time is also shown, taking into account the fact that each statistic had to exceed its respective threshold in 11 consecutive samples in order to detect the fault. This procedure is applied in order to avoid false alarms.

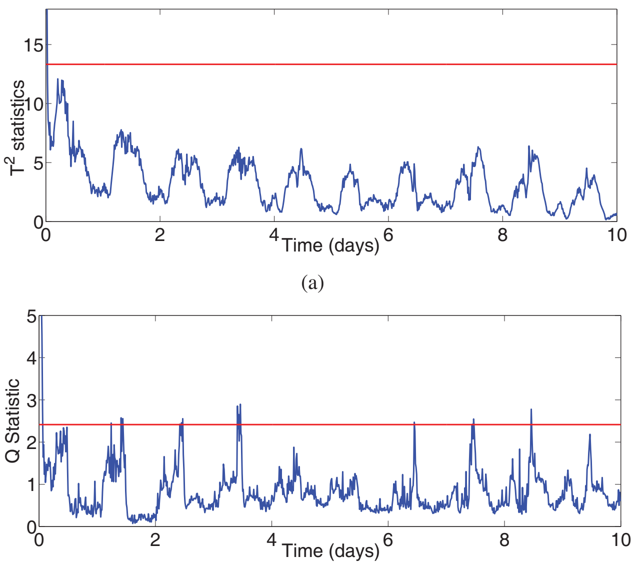
Figure 3 shows the results when the toxicity shock fault with a magnitude of 20% occurred at the second day, i.e., $t = 200$ sampling times, tested with the T^2 (a) and Q (b) charts, respectively, with the classical PCA monitoring scheme. As the figure shows, this scheme cannot detect the fault. However, in Figure 4, the results of the same fault are shown with the NNPCA monitoring scheme explained in this paper. The fault is detected rapidly, in the same instant it occurs, as the figure shows. Figure 5 also shows the contribution plot for this fault and, as it is possible to see, the residual responsible for this fault is the residual corresponding to variable $X_{B,H}$, i.e., the fault is well diagnosed.

The NNPCA scheme is able to detect the first two faults for all sizes with a very short detection time, as Table II

TABLE II

DETECTION TIME FOR THE NNPCA AND PCA METHODS

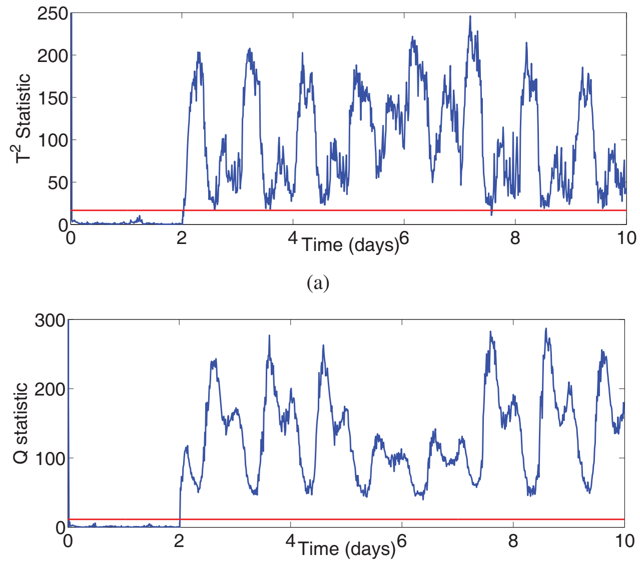
Fault	Size	NNPCA detection time		PCA detection time	
		T^2	Q	T^2	Q
Toxicity shock	10%	214	222	–	–
	20%	216	216	–	–
	40%	214	214	–	353
	60%	214	214	267	215
	80%	214	214	214	214
Inhabitation	10%	214	222	–	–
	20%	216	216	–	–
	40%	214	214	–	221
	60%	214	214	262	215
	80%	214	214	222	214



(a)

(b)

Fig. 3. The T^2 (a) statistic and the Q (b) statistic when a toxicity shock fault with a magnitude of 20% occurs at time instant 200 using the PCA monitoring scheme



(a)

(b)

Fig. 4. The T^2 (a) statistic and the Q (b) statistic when a toxicity shock fault with a magnitude of 20% occurs at time instant 200 using the NNPCA monitoring scheme

shows. The PCA monitoring scheme is only able to detect faults with the Q control chart for faults bigger than 40% in magnitude. As an example, in Fig. 6, the results for the inhabitation fault with a magnitude of 40% using the classical PCA are shown and in the Fig. 7 the same results are shown for the NNPCA method. The T^2 statistic is not able to detect the fault in the classical PCA method, and the Q statistic detects the fault after 21 sampling times. The detection time for the same statistics in the NNPCA method is $t = 214$, i.e., 14 sampling times after the occurrence of the fault, taking into account the fact that the statistic in both methods has to exceed the threshold in 11 consecutive sampling times to detect the fault.

However, the third fault is very difficult to detect, i.e., both methods can only detect this fault with a magnitude over 90%. This is because this fault occurs in the settler and the variables used in the monitoring scheme are collected at the output of the biological treatment, and they are not very affected by this type of fault. In order to improve the detection of this fault, variables related with the settler have

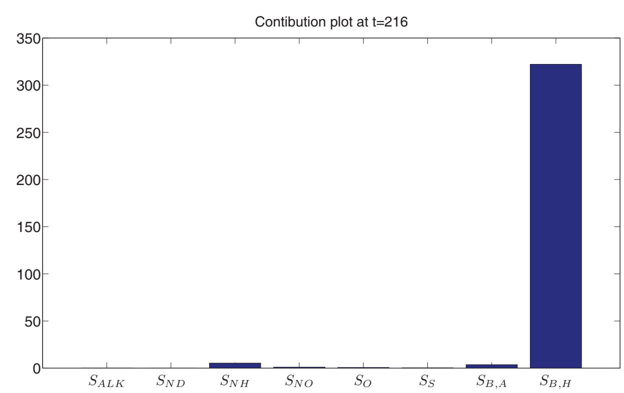
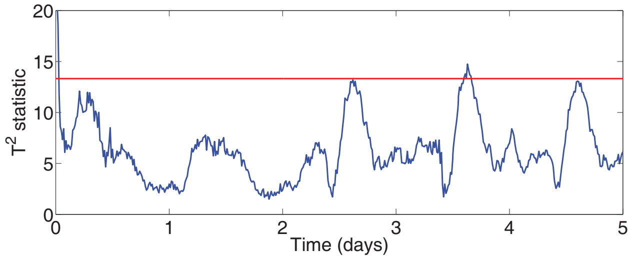
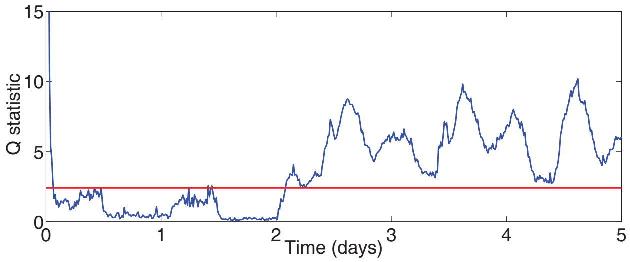


Fig. 5. Contributions plot of the normalized error of the variables for a 20% toxicity shock fault

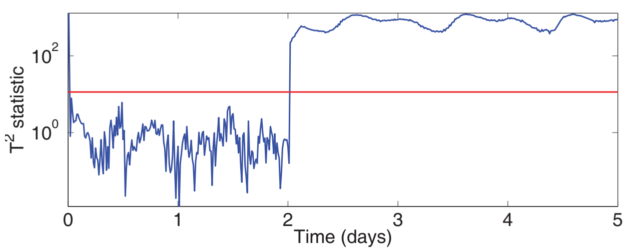


(a)

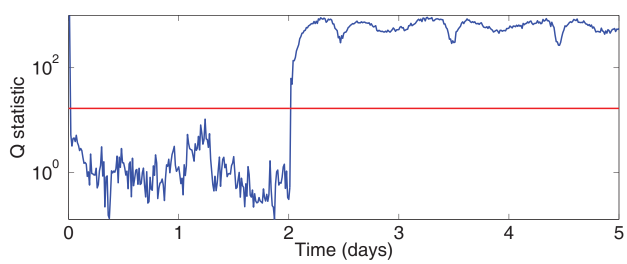


(b)

Fig. 6. The T^2 (a) statistic and the Q (b) statistic when a inhabitation fault with a magnitude of 40% occurs at time instant 200 using the PCA monitoring scheme



(a)



(b)

Fig. 7. The T^2 (a) statistic and the Q (b) statistic when a inhabitation fault with a magnitude of 40% occurs at time instant 200 using the NNPCA monitoring scheme

to be used in the monitoring process. For this reason, this fault does not appear in Table II.

IV. Conclusions

In this paper, an NNPCA method that integrates neural networks and principal component analysis is used to detect faults in a wastewater treatment plant. The neural network is used to learn the normal dynamic system behavior based on operating data. Residual generation is carried out as the difference between the actual behavior of the system and the predicted output given by the neural network driven by the same inputs. PCA is used to evaluate the residual variables, generating monitoring charts. It can evaluate the current performance of the process and detects the faults.

This technique has been applied to the simulation of a benchmark of the biological wastewater treatment process, a highly non-linear process. Three different faults are generated in this plant and the NNPCA method is compared with the classical PCA technique. The results show that the NNPCA method is more effective than the classical PCA, and this NNPCA method can increase the sensitivity and robustness of the monitoring schemes, i.e., it is able to detect faults with a small magnitude.

V. Acknowledgments

This work was supported by the Spanish Ministry of Science and Innovation under project DPI2009-14410-C02-02.

References

[1] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, and K. Yin, "A review of process fault detection and diagnosis. Part I: Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, pp. 291–311, 2003a.

[2] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, and K. Yin, "A review of process fault detection and diagnosis. Part II: Qualitative models and search strategies," *Computers & Chemical Engineering*, vol. 27, pp. 313–326, 2003b.

[3] V. Venkatasubramanian, R. Rengaswamy, and S. Kavur, "A review of process fault detection and diagnosis. Part III: Process history based methods," *Computers & Chemical Engineering*, vol. 27, pp. 327–346, 2003c.

[4] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker Inc., 1998.

[5] J. Chen and R. J. Patton, *Robust Model-Based Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

[6] T. Kourti, "Process Analysis and Abnormal Situation Detection: From Theory to Practice," *IEEE Control System Magazine*, pp. 10–25, 2002.

[7] B. Wise and N. Gallagher, "The Process Chemometrics Approach to Process Monitoring and Fault Detection," *Journal of Process Control*, vol. 6, pp. 329–348, 1996.

[8] J. Korbicz, J. Koscielny, Z. Kowalczuk, and W. Cholewa, *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, 2004.

[9] W. Ku, R. Storer, and C. Georgakis, "Disturbance detection and isolation by dynamic principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 30, pp. 179–196, 1995.

[10] A. Simoglou, E. Martin, and A. Morris, "Statisical performance monitoring of dynamic multivariate processes using state space modeling," *Computers and Chemical Engineering,*, vol. 26(6), pp. 909–920, 2002.

[11] M. Kramer, "Autoassociative Neural Networks," *Computers & Chemical Engineering*, vol. 16, pp. 313–328, 1992.

[12] D. D and T. McAvoy, "Nonlinear principal component analysis based on principal curves and neural networks," *Computers & Chemical Engineering*, vol. 20, pp. 65–78, 1996.

[13] C. Cheng and M.-S. Chiu, "Nonlinear process monitoring using jitl-pca," *Chemometrics and intelligent laboratory systems*, vol. 76, pp. 1–13, 2005.

[14] T. Kourti and J. MacGregor, "Multivariate SPC Methods for Process and Product Monitoring," *Journal of Quality Technology*, vol. 28, pp. 409–428, 1996.

[15] O. Nelles, *Nonlinear system identification*. Springer, 2001.

[16] L. Chiang, E. Russell, and R. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. Springer, 2000.

[17] A. Ferrer, "Multivariate Statistical Process Control Based on Principal Component Analysis (MSPC-PCA): Some Reflections and a Case Study in an Autobody Assembly Process," *Quality Engineering*, vol. 19, pp. 311–325, Oct. 2007.

[18] *The COST Simulation Benchmark: Description and Simulator Manual*.