Available online at www.sciencedirect.com

SCIENCE DIRECT®

Analytica Chimica Acta 515 (2004) 101–108

ANALYTICA CHIMICA ACTA

www.elsevier.com/locate/aca

Multivariate monitoring of fermentation processes with non-linear modelling methods

J.A. Lopes*, J.C. Menezes

Center for Biological & Chemical Engineering, Technical University of Lisbon, Av. Rovisco Pais, P-1049-001, Lisbon, Portugal

Received 8 September 2003; received in revised form 13 January 2004; accepted 20 January 2004

Available online 14 April 2004

Abstract

Multiway principal components analysis (MPCA) and parallel factor analysis (PARAFAC) are widely used in exploratory data analysis and multivariate statistical process control (MSPC). These models are linear in nature, thus, limited when non-linear relations are present in the data. Principal component analysis (PCA) can be extended to non-linear principal components analysis using autoassociative neural networks. In this paper, the network's bottleneck layer outputs (non-linear components) were made orthogonal. A method to estimate confidence limits based on a kernel probability density function was proposed since these limits do not assume that the non-linear scores are normally distributed. A measure for the non-linear scores (D_{NL}) was presented here to monitor on-line the process replacing the well known Hotelling's T^2 statistic. One hundred and two industrial fermentation runs were used to evaluate the performance of a non-linear technique for multivariate process statistical monitoring. Three process runs with faults were used to compare the error detection performance using a statistic for the non-linear scores and the residuals statistic (SPE).

Keywords: Exploratory data analysis; Statistical process control; MPCA; Neural networks; Kernel density estimation

1. Introduction

Dynamic models for industrial fermentation processes are difficult to identify because of a wide variety of reasons: microorganisms complex dynamics, variable and ill-defined raw materials, dependence on previous process stages (strain selection and pre-culture production) [1]. The performance of a fermentation is often related to the amount of product obtained at the end of process [2]. Therefore, it is essential to ensure that each production run is as close as possible to the set of pre-established optimal process trajectories.

The idea of statistical process control (SPC) is that it is possible to capture the nominal operation process state based on historical process information [3,4]. It is based on univariate process charts which are traditionally used to monitor the process. In multivariate statistical process control (MSPC) the correlation between the original variables is considered, which decreases the number of false alarms. Batch MSPC is the extension of MSPC to batch processes. Several statistical tools such as principal component analysis (PCA) are commonly used to explore historical data. The aim of statistical projection techniques is to create from the process monitored variables a new set of variables (the principal components or scores) which reflects the true problem dimensionality [5,6]. Because there are three modes in batch process data (runs, variables and time), two-dimensions must collapse in one-dimension in order to apply PCA (multiway principal components analysis, MPCA) [7]. Nomikos and MacGregor [6] describe three different ways to analyse processes on-line using MPCA models.

Several authors suggested the application of auto-associative neural networks [8,9]. These special type of networks are able to project data in a lower dimensional space in much the same way as PCA. These projections (the non-linear scores) are non-linear and capture more process data variance than if obtained with a linear method.

Multivariate statistical control charts are based on the scores (Hotelling's T^2 statistic) and the residuals (squared prediction error, SPE statistic or Q). Hotelling's T^2 statistic measures the variability explained by the model, while the SPE statistic measures the non-explained variance [5]. Confidence limits for the Hotelling's T^2 statistic are based

* Corresponding author. Tel.: +351-218-419-167; fax: +351-218-419-197.

E-mail address: bsel@ist.utl.pt (J.A. Lopes).

0003-2670/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.aca.2004.01.060

on the Fisher's F-distribution which assumes that scores are normally distributed. Confidence limits for the SPE statistic are based on the chi-square distribution [5].

When the true distribution is unknown, a non-parametric method must be used to estimate it. Kernel functions (e.g., based on gaussian functions) can be used to this purpose. If the distribution is based on a density estimation method, a measure needs to be developed to replace Hotelling's T^2. Martin and Morris described a method (M^2 statistic) to generate confidence limits based on a density estimator [10]. We hereby define a measure D_{NL} to replace the Hotelling T^2 in order to use the non-linear scores to monitor the process.

When multivariate methods are used to monitor a process, the diagnosis of a fault is based on the estimation of the contributions of each original variable to the scores statistic (Hotelling T^2 in the case of a linear model) and error statistic (SPE). These contributions are compared with the average contributions obtained for the nominal data to detect the problem origin.

2. Theory

The MPCA is a linear model based on a data decomposition commonly named unfolding to transform a three-way data array $\underline{X}$ (run × variables × time) in a two-way array X (run × variables × time) where in general the run dimension is preserved. Using singular value decomposition, it is possible to extract an array T (runs × components) of scores (variability among runs) and a loading matrix $\boldsymbol{P}$ (variables × time × components) containing information on variables and time dimensions. E (runs × variables × time) is the residuals array that depends on the chosen number of components [6]. The PCA model is given by Eq. (1):

$$X = T\boldsymbol{P}^{\mathrm{T}} + E \tag{1}$$

Scores for new observations, $\hat{x}$ (1×variables), are obtained using the MPCA loadings $\boldsymbol{P}$. Note that observations are defined as row vectors.

$$\hat{t} = \hat{x} \times \boldsymbol{P} \tag{2}$$

In Eq. (2), $\hat{t}$ are the scores (1×components) corresponding to the new observation $\hat{x}$.

A non-linear principal component, results from the projection of a set of variables in a principal curve [11]. These curves can be defined by any linear or non-linear function (e.g., a neural network). Jia et al. [12] described the extraction of non-linear principal components with a neural network.

The autoassociative neural network is a static feedforward neural network with a structure adjusted for the compression of information [9]. Compression is achieved through the inclusion of an interior bottleneck layer. In an autoassociative network, inputs and outputs are equal. Therefore, information is compressed and decompressed as it flows through the network. Typically these networks have five layers (see Fig. 1). The size of the first and the last layers is defined by the number of inputs. The bottleneck layer determines the number of non-linear principal components required. In the second and fourth layers (mapping and demapping layers), non-linear activation functions are generally used. Linear functions are used in the remaining layers.

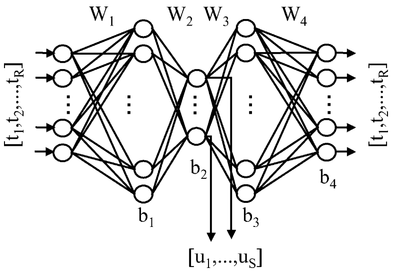

Fig. 1. Five layer autoassociative neural network topology used for non-linear principal component analysis. Non-linear scores (u) are obtained after the bottleneck layer (W: weights; b: bias; t: linear scores; u: non-linear scores).

The criteria to train a feedforward network is the minimization of a function (typically the sum of squared errors). However, this approach must be extended in the context of this work because this does not forces bottleneck vectors to be orthogonal. The aim is to force the internal layer outputs (non-linear scores) to be orthogonal. Ensuring orthogonality prevents somehow the network to overfit the data and provides more reproductible results for different network initializations. A modification of the training method is thus required. Eq. (5) shows the two terms of the new objective function F.

$$F = \eta\mathrm{SSE} + (1-\eta)\Phi \tag{3}$$

In Eq. (3), the term SSE is the sum of squared errors (measured at the output layer) and the second term is a measure of the colinearity among the non-linear scores vectors (Eq. (4)). Note that increasing the η parameter towards 1 will result in the usual mean squared error (MSE) minimization. We found in this case that setting η equal to 0.5 was enough to ensure a correlation coefficient between non-linear scores lower than approximately 0.01.

$$\Phi = \sum_{i=1}^{s}\sum_{j=i+1}^{s}\frac{S_{ij}}{\sqrt{S_i S_j}} \tag{4}$$

In Eq. (4), S_{i} and S_j are the scores i and j variances, respectively, and S_{ij} the covariance between the scores and S is the number of non-linear scores.

If the data is normally distributed, then the linear scores (which are linear combinations of the variables), by the central limit theorem, should be approximately normally distributed. Confidence limits for the Hotelling's T^2 statistic are thus based on Fisher's F-distribution. The Hotelling T^2 will not be used in this work.The SPE statistic for a time instant i (SPE$_i$) is given by Eq. (5) where $\boldsymbol{X}$ is the data matrix, $\hat{\boldsymbol{X}}$ the model estimated data matrix, i the sample index and j the variable index.

$$SPE_i = \sum_{j=1}^{J} (X_{ij} - \hat{X}_{ij})^2 \tag{5}$$

Confidence limits for SPE statistic are obtained using Box's approximation [3,7]. Westerhuis et al. [13] explain how to compute robust limits for the SPE statistic. Confidence limits are estimated by building several models, each one, leaving one sample out. The errors computed for these unused samples is then used to estimate the confidence bounds (note that the SPE statistic follows a $g\chi_{\mathrm{h}}^2$ distribution with appropriate degrees of freedom).

When non-linear scores are used in a statistical control chart one cannot consider that the data arose from a normal distribution. A density estimator based on a kernel density function was used to estimate the real population distribution. The result is a contour if two scores are used, a surface if three scores are used and so on. Interior points were discovered with the so-called odd-parity concept [14]. This states that a point is interior if the number of intersections between the boundary and a straight line connecting this point with an exterior point is odd.

We define a measure D_{NL} to follow the process based on the s-dimensional non-linear score vectors and the estimated contour confidence limit. The measure D_{NL} is defined as a linear function of the distance in the s-dimensional space between a projected non-linear score and the closest point on the confidence limit boundary (d). It is convenient in order to interpret the statistical control chart to define a value below which the process is in-control. An easy way to achieve this uses a parameter MID defined as the maximum value the distance d can take when the projected non-linear score is inside the boundary. D_{NL} is obtained with Eq. (6). To compute D_{NL} it is necessary to know if the projected score is in or out-of-control ($c = 1$ if in-control and $c = -1$ if out-of control).

$$D_{\mathrm{NL}} = 1 - c\frac{d}{\mathrm{MID}} \tag{6}$$

The value for this statistic depends on the confidence limit chosen for the boundary. Thus, the confidence limit for this statistic at a significance level α is always the value $D_{\mathrm{NL},\alpha} = 1$. D_{NL} is below 1, if the process is in-control and above 1 if otherwise.

After a new sample projection, if an error is detected (either in the D_{NL} or SPE statistic) the error source must be investigated. Contributions to non-linear scores are more difficult to obtain than contributions to the linear scores since the network is a non-linear model. Here, we compute at each time point the derivatives of each non-linear score (u_j) in order to each input variable (x_i): $\partial u_j/\partial x_i$. These derivatives are then compared to the ones obtained for the nominal samples to assess for error sources. Here, diagnostic is obtained directly from the non-linear scores and not from the D_{NL} statistic-like itself. How to compute these derivatives is beyond the scope of this paper, but the reader can find the solution in [15].

Contributions for the SPE statistic are computed much in the same way they are for a linear model. Considering $\boldsymbol{x}$ the vector containing the new observation and $\hat{\boldsymbol{x}}$ the corresponding network output, the contribution of variable i for the SPE statistic is given by Eq. (7).

$$C_{\mathrm{SPE}}(\boldsymbol{x}_i) = \frac{(x_i - \hat{x}_i)^2}{Q_i} \times 100 \tag{7}$$

The 95% confidence bounds for partial derivatives of non-linear scores and SPE contributions were used in the charts to assess for significative deviations from nominal values when projecting new data. These confidence bounds were calculated for each time point separately. The procedure for each time point was as follows: (1) the derivatives and SPE statistic were obtained for each nominal run leaving that run out of calibration data and (2) a 95% confidence interval is then estimated for each variable. For this process, the 95% confidence intervals for partial derivatives and SPE statistic are stored in matrices with dimensions 24 × 7 × 2 (time points × variables × low value/high value).

3. Experimental

The process used in this work was the industrial production by fermentation of an antibiotic-like active ingredient product (API) [16]. Fed-batch cultivation of a *Streptomycetes* strain was carried out using a non-defined (complex) medium containing soy meal and a carbon source. The conditions used were typical of those employed routinely in industry for aerobic microbial growth [17]. Inocula were produced on a 100 dm^3 pre-culture tank. The fermentation process lasts approximately 140 h. After the production fermentation step the culture media was transferred for downstream processing. Seven variables from the liquid phase were monitored and sampled with a frequency of about 6 h. Table 1 lists the variables used throughout this work. Data pre-processing included outliers detection and

Table 1
List of the variables in the dataset

	Variable	Description
1	pH	Fermentation media pH
2	DO	Concentration of dissolved oxygen in the fermentation media
3	PMV	Packed mycelial volume in fermentation media
4	[C]	Defined carbon source concentration in the fermentation media
5	[S]	Starch concentration in the fermentation media
6	[N]	Inorganic nitrogen concentration in the fermentation media
7	[API]	Active product ingredient concentration in the fermentation media

Table 2
Description of the faulty runs

Runs	Problem	Event time (h)
A	Different quality of raw-material used	0
B	Problem on pH (too low during production phase)	43
C	Error on the defined carbon source feeding rate	45

noise reduction. All variables were mean centred (removing the time mean profile over the runs).

The data set contains 102 runs operated under nominal operating conditions (NOC). The dimensions of the array containing the data were (102 × 7 × 24) (runs × variables × time). Three faulty runs were used to test the D_{NL} based monitoring approach (runs A–C). Table 2 describes the problems on these runs. These three runs were part of an experimental planning carried out at the plant, thus, the changes were in did intentional, although in this context they have been treated as faults.

4. Results and discussion

The number of nodes to use in the network's second and fourth layers was optimized by a strategy based on cross-validation. For simplicity reasons, the number of nodes in the second and fourth layers was assumed to be equal. The original dataset is divided randomly in two sets: training (70% of the data) and testing (30% of the data) sets. After training the network with the training set a prediction is obtained for the testing set, being stored the corresponding mean squared error. Because this process is dependent on the training/testing split and network parameter initialization, it was performed 20 times for each combination of hidden and second/fourth layer nodes. So, each network topology presented in the remaining of the section represents the optimum obtained by this process.

Several autoassociative networks (with 1–4 hidden nodes) were generated to assess the advantage of using the non-linear model to compress data. The 102 nominal runs were unfolded yielding a matrix $\boldsymbol{X}$ with dimensions 102 × 168 (runs × time × variables) where each row corresponds to a run. A MPCA with five linear components was applied resulting in a scores matrix with dimensions 102 × 5 (70.7% of captured variance). This matrix was used to obtain the non-linear components. The original data (dimensions 102 × 168) was not used directly because the number of network input/output nodes (168) will be too high (resulting in too many network parameters). Because the training is dependent on the network initialization, 50 networks with the same topology were trained and the best result (in terms of training fitting error) was stored. The same number of non-linear components captures more variance than the linear components as expected. Table 3 compares the captured variance for MPCA and non-linear models. The number of second/fourth layer nodes for each number of hidden nodes was selected using the optimization criteria depicted above. For example, the percent captured variance in two MPCA components was 43.5% while for two non-linear components the global captured variance was 59.1%. Note that for five non-linear components, the captured variance is equal to the variance captured in five linear MPCA components.

Table 3
Comparison between cumulative captured variance in linear components and non-linear components

Hidden nodes (components, *s*)	Second/fourth layer nodes (*t*)	MPCA (%)	Neural network (%)
1	3	24.9	42.3
2	4	43.5	59.1
3	4	57.1	65.9
4	4	64.7	69.2
5	4	70.7	70.7

The non-linear components were extracted using five MPCA linear components with an autoassociative network with topology 5-*t*-*s*-*t*-5.

The two non-linear component model was selected to compare the performance of the two scores confidence bounds approaches (parametric and density estima tion). The faulty runs were projected in the MPCA model (to obtain the linear scores) and these were further projected in the autoassociative network (obtaining the non-linear scores). These non-linear scores were compared with the nominal non-linear scores. The plots in Fig. 2 show the 95 and 99% confidence limits obtained with the parametric statistic (left) and with the density estimation method (right). In the first case, two non-nominal runs are inside the 99% confidence limit. In the second case, all three runs were detected as outliers. This shows the advantage of the density estimation method over the parametric statistic.

However, this approach is not adequated for on-line monitoring since the entire run must be known. Nomikos and MacGregor [6] present three methods to monitor on-line a batch process using a single model. Another method could be based on a sliding window, but this involves the use of several models. Here, we tried to use the nominal runs stacked on top of each other to build only one model (each row is a time point, thus, no unfolding required). Note that the covariance structure changes over time in such a process, but since a non-linear model is being used these changes might be better captured than with a linear model.

In this case, no previous linear model was built, since the original data was used to train the network (only seven process variables were available which means than only seven input/output nodes are required). Therefore, the data for the 102 nominal runs was arranged by stacking one run on top of each other yielding a matrix $\boldsymbol{X}$ with dimensions 2448 × 7. The later was used to obtain the neural network model as explained in Section 2 (7-4-2-4-7 architecture). The training

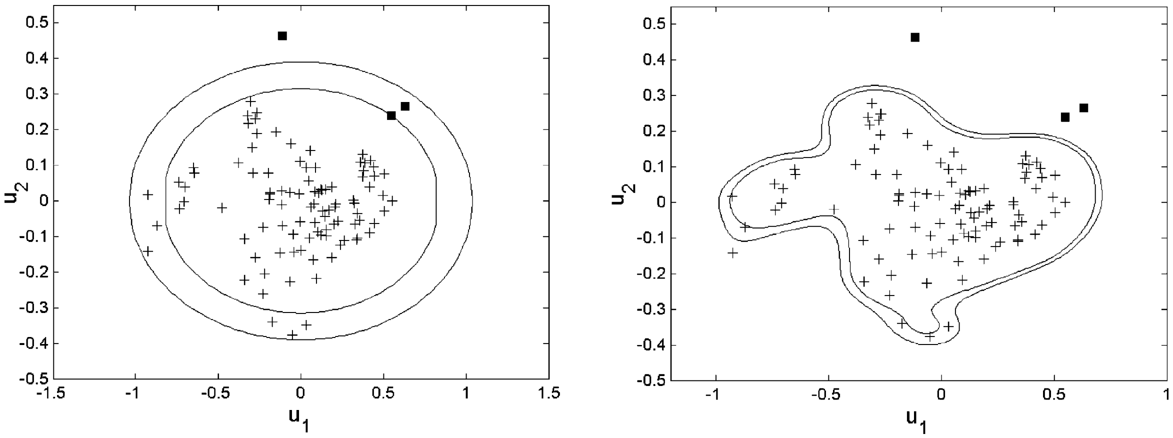

Fig. 2. Non-linear scores obtained for the 102 nominal runs (+) where each score represents a run (multiway approach). Three faulty runs (■) were projected on the model (left: confidence limits assuming normal distribution, 95 and 99%; right: confidence limits based on the density estimation method, 95 and 99%).

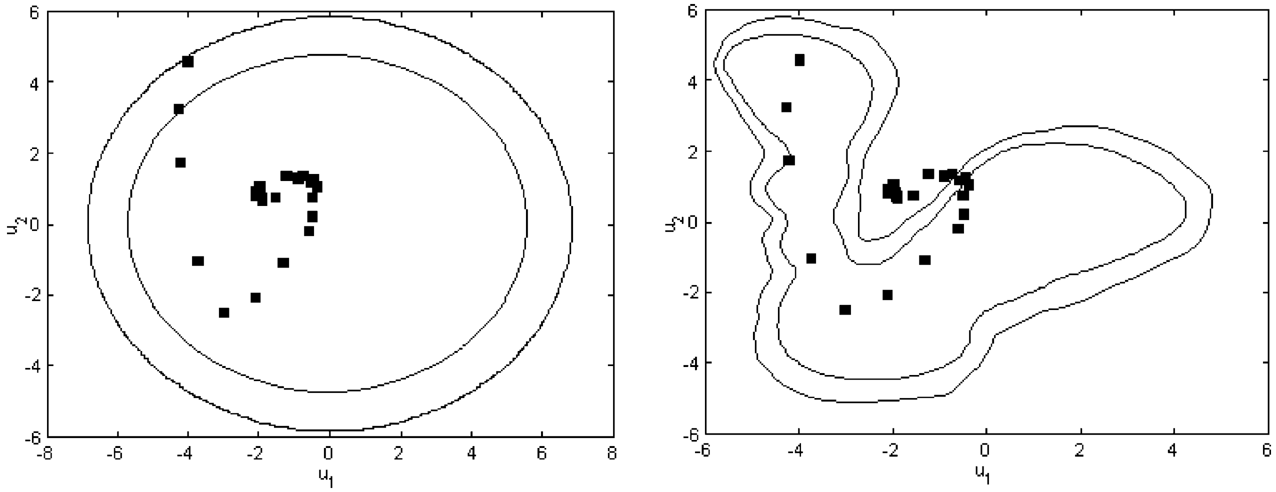

Fig. 3. Non-linear scores extracted for the 102 nominal runs where each score represents a time point. The faulty run B was projected (■) while points relative to nominal runs were not shown for simplicity (left: confidence limits assuming normal distribution, 95 and 99%; right: confidence limits based on the density estimation method, 95 and 99%).

predicted variance for this model was 63.5%. As before, the three faulty runs were projected on the network yielding the corresponding non-linear scores.

Fig. 3 shows the confidence limits obtained with the parametric and non-parametric approaches. The scores for the faulty run B were projected. The scores are not normally distributed as can be seen by the shape of the contours obtained with the density estimation method. In this case, the fault could be observed. Once again, the density estimation method proved to be more reliable.

The proposed D_{NL} values were obtained for the three faulty runs and also for a nominal run (control). D_{NL} values were obtained considering the 95% confidence bound. In the chart, the boundary value corresponding to the 95% confidence region is always the value 1. Fig. 4 shows the multivariate statistical process control chart based on this value. The nominal run remained in-control during the entire run (D_{NL} always below 1 as expected).

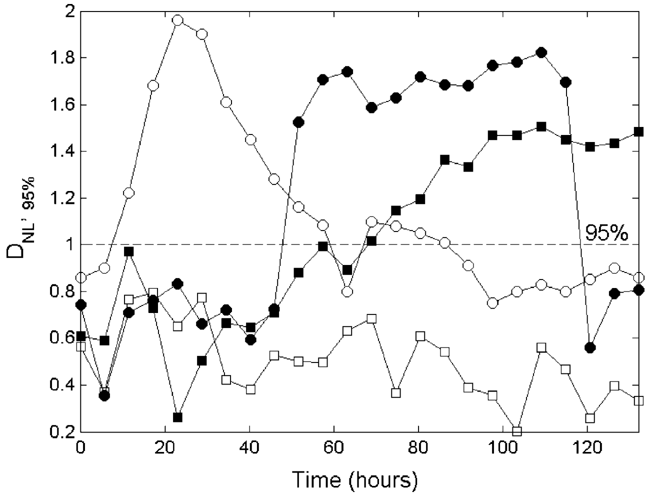

Fig. 4. D_{NL} statistical process control chart obtained using 95% confidence limits based on a density estimation method ((□) selected nominal run; (○) run A; (■) run B; (●) run C).

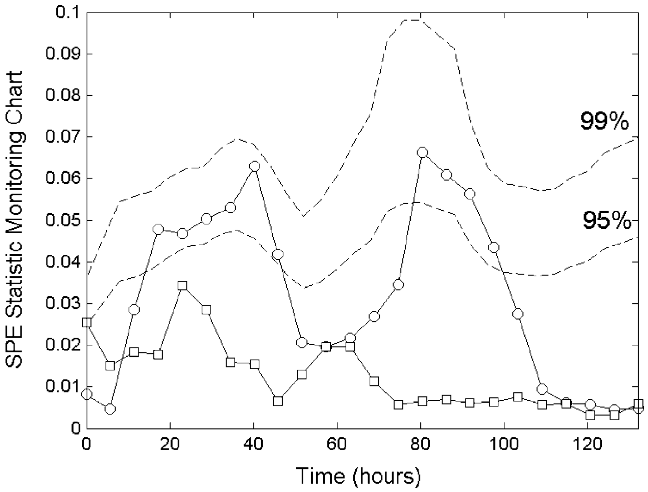

Fig. 5. Squared prediction error (SPE) statistical monitoring chart ((□) selected nominal run; (○) run A; (- - -) confidence limits).

The effect of the different raw-material quality used at the beginning of run A was detected after 8 h. After about 60 h, the process returned to its normal state mainly because the effect of the different raw-material quality was more evident throughout the microrganism growth phase (between 10 and 40 h).

The fault in run B occurred after 43 h, but it was only detected after 52 h in the D_{NL} chart. This was observed because the effect of a pH decrease after the production phase is cumulative. Therefore, the data reflected it only after about 10 h. This fault was not corrected, thus, the D_{NL} values remained above the confidence limits towards the fermentation end. Note that the information provided by the D_{NL} values for run B was already visible in the second chart of Fig. 3.The problem in run C was immediately detected after the occurrence since the change in the carbon source feed rate was quite severe. In this case, the process returned to a controlled state after 120 h because the feeding was stopped.

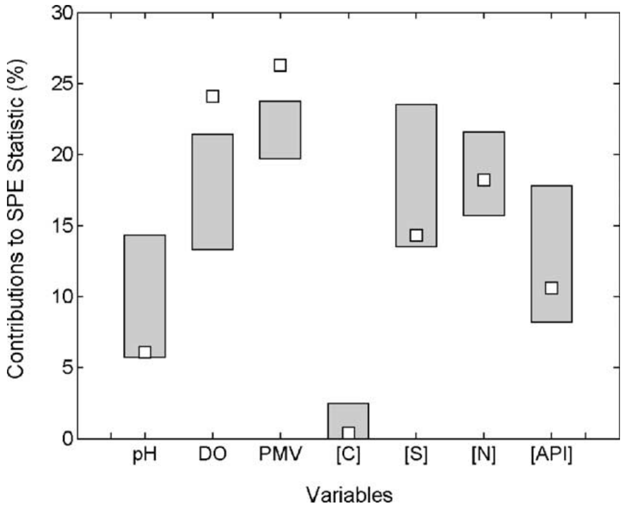

Fig. 7. Contributions of original variables to the quared prediction error (SPE) statistic for run A at 18 h ((□) contributions for run A at 18 h; (▨) nominal contributions estimated with model training data, 95%).

The D_{NL} statistical chart must be complemented with the SPE chart. Fig. 5 shows the SPE statistical monitoring chart for the control experience already depicted in Fig. 4 and for run A (different soybean flour used). The statistical chart signals a problem in two regions: between 18 and 48 h and between 80 and 100 h. The first region contains the microorganism growing phase and the beginning of API production phase.

When the statistical charts signal a possible problem, the fault diagnosis strategy should be fired to investigate the causes. To illustrate the fault diagnosis methodology we selected run A at 18 h to be diagnosed. Partial derivatives for this run at 18 h were calculated for both non-linear scores and compared with the nominal regions. Fig. 6 depicts these results. For the second non-linear component, nothing appears to be different from the normal situation. However, for the first non-linear component the derivatives for pH

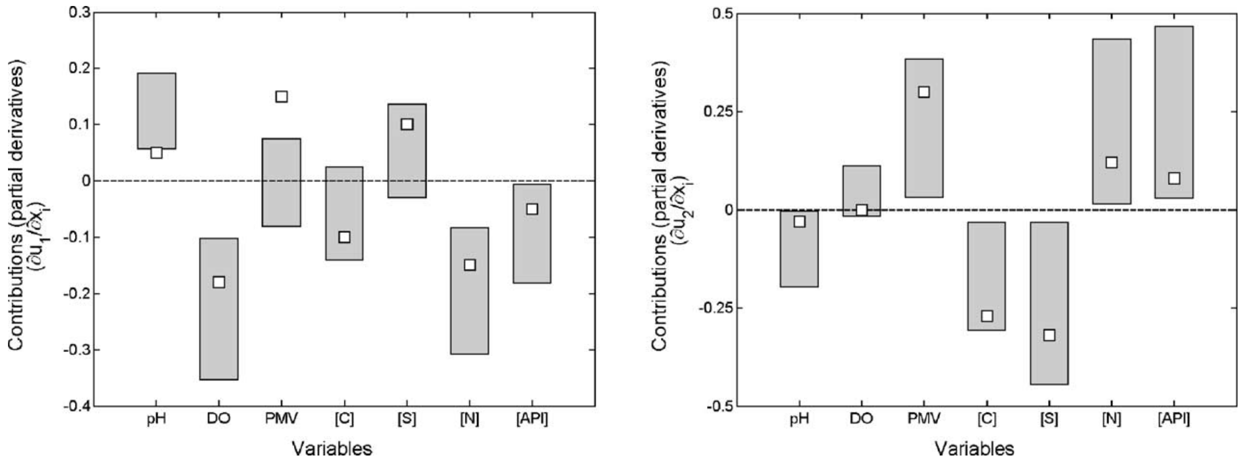

Fig. 6. Partial derivatives of non-linear components 1 (left) and 2 (right) in order to each original variable for faulty run A at 18 h ((□) partial derivatives for run A; (▨) nominal partial derivatives region estimated with training data, 95%).

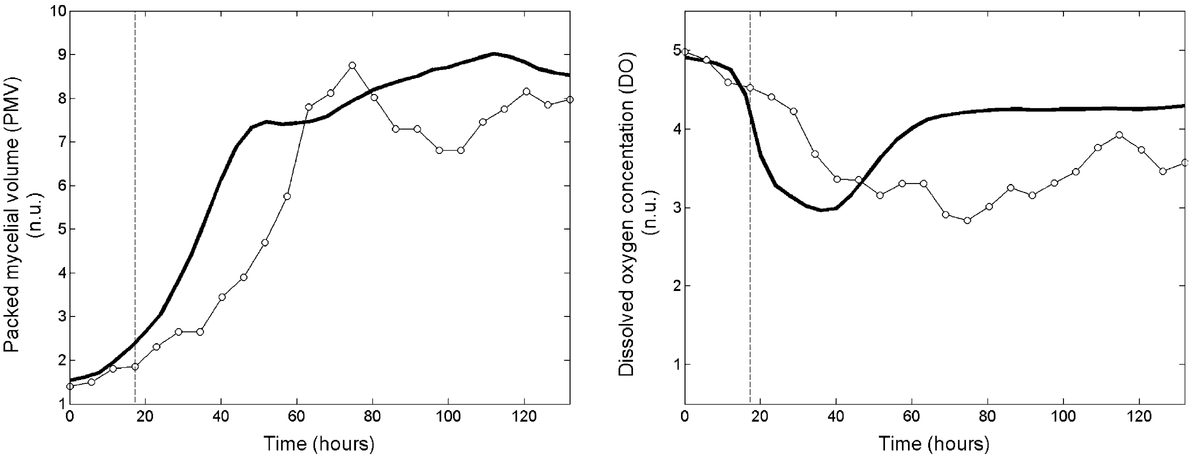

Fig. 8. Packed mycelial volume (left) and dissolved oxygen concentration (right) profiles for faulty run A (○) and nominal runs averaged profiles (—). The data is shown in normalized units. A marker at 18 h is shown (- - -).

and packed mycelial volume (PMV) were outside the corresponding 95% normal region. A similar procedure was performed for the SPE statistic. Fig. 7 contains the obtained results. Here, contributions for dissolved oxygen (DO) and packed mycelial volume are outside the confidence limits. So, the reasons for the possible fault after the statistical charts would be pH, DO and PMV. Fig. 8 shows the time profiles for DO and PMV for run A. These were compared with the nominal time profiles. The later were obtained by averaging each profile from the 102 runs of the nominal dataset. A late microorganism growing can be identified by inspecting the PMV profile. Although the maximum biomass concentration is approximately the same for run A and for an ideal run, this value happens with a 20 h delay. This delay is also present in the DO time profile. Because there is less biomass in the beginning of the fermentation, the dissolved oxygen concentration remains higher for a longer period. After 40 h, the dissolved oxygen concentration remained low in comparison with the ideal data until the end of the run.

5. Conclusions

In this paper, the linear model PCA was complemented with an autoassociative neural network model in order to generate non-linear principal components. These methods were used to analyse an industrial fermentation process. Neural networks outperformed the linear model in terms of percentage of captured variance for the same number of components.

In general, the advantage of the non-linear model decreases as the number of non-linear components is increased. The advantage of the non-linear model will depend on the number of nodes used in the second and fourth layers. In this paper, we used four nodes in these layers and two nodes in the hidden layer.

With the process data used throughout this work, the non-parametric density estimation method (based on kernel functions) presented an advantage over the parametric statistic to generate confidence limits for the scores since they are no normally distributed (the parametric statistic assumes that data is normally distributed which was not the case).

A statistic-like parameter (D_{NL}) was proposed to evaluate on-line scores for new runs using the density estimated confidence bounds and complemented with the squared prediction error statistic. Three faulty runs were monitored using these multivariate statistics. The fault diagnosis methods were tested to diagnose a fault ocurred in one faulty run. They provided acurate information about the error sources which in turn, could be biochemically explained.

Many theoretical aspects concerning non-linear components extraction need to be addressed. Implementation of orthogonal functions in the autoassociative neural network bottleneck layer might optimise the non-linear components extraction. Improvements are also required in the network's training algorithm and structure fine-tuning.

Acknowledgements

The authors would like to thank Companhia Industrial Produtora de Antibióticos SA (CIPAN) in Portugal for providing the data. The financial support from the Foundation for Science and Technology (BPD/7194/2001) is gratefully acknowledged.

References

[1] B. Atkinson, F. Mavituna, Biochemical Engineering and Biotechnology Handbook, Stockton Press, London, 1992, p. 870.
[2] J. Lopes, J. Menezes, AIChE Symp. Series 94 (320) (1998) 391.
[3] B. Wise, N. Gallagher, J. Proc. Cont. 8 (6) (1996) 329.
[4] G. Chen, J. McAvoy, J. Proc. Cont. 8 (5) (1998) 409.
[5] B. Wise, N. Gallagher, S. Butler, D. White, G. Barna, J. Chemom. 13 (1999) 379.
[6] P. Nomikos, J. MacGregor, Technometrics 1 (37) (1995) 41.

[7] E. Martin, J. Morris, Trans. Inst. MC 18 (1) (1996) 51.

[8] U. Thissen, W. Melssen, L. Buydens, Anal. Chim. Acta 446 (2001) 371.

[9] R. Kocjancic, J. Zupan, J. Chem. Inf. Comput. Sci. 37 (1997) 985.

[10] E. Martin, A. Morris, J. Proc. Cont. 6 (6) (1996) 349.

[11] E. Malthouse, A. Tamhane, R. Mah, Comput. Chem. Eng. 21 (8) (1997) 875.

[12] F. Jia, E. Martin, A. Morris, Comput. Chem. Eng. 22 (1998) 851.

[13] J. Westerhuis, S. Gurden, A. Smilde, J. Chemom. 14 (1999) 335.

[14] J. Foley, A. Dam, S. Feiner, J. Hughes, in: C. Reading (Eds.), Computer Graphics: Principles and Practice, Addison-Wesley Publishing Company, New York, p. 1175.

[15] M. Hagan, H. Demuth, M. Beale, Neural Network Design, PWS Publishing, Boston, 1996.

[16] A. Neves, L. Vieira, J. Menezes, Biotechnol. Bioeng. 72 (6) (2001) 628.

[17] W. Strohl, Biotechnology of Antibiotics, second ed., Marcel Dekker Inc., New York, 1997, p. 432.