A Batch-Incremental Process Fault Detection and Diagnosis Using Mixtures of Probablistic PCA

Thiago Nakamura
Electronic Engineering Departament
Federal Univeristy of Minas Gerais
Av Antonio Carlos, 6627 - Belo Horizonte - Brazil
Email: akionakamura@ufmg.br

Andre Lemos
Electronic Engineering Departament
Federal Univeristy of Minas Gerais
Av Antonio Carlos, 6627 - Belo Horizonte - Brazil
Email: andrepl@cpdee.ufmg.br

***Abstract*—In process engineering, a fast and efficient fault detection and diagnosis (FDD) system is an essential component to improve both safety and productivity losses under abnormal conditions. Over the years, techniques based on models derived from process historical data, specially under a probabilistic framework, have gain a lot of attention. In this paper, probabilistic principal component analysis (PPCA) mixture models are used to cope with the FDD task. A batch-incremental method is proposed for statistical process monitoring, seeking to detect and learn new faulty behaviour, or yet, diagnose an already known fault. The proposed methodology was applied to the Tennessee Eastman Process under a closed-loop control, and it has shown robust and reliable results.**

I. Introduction

Every year the industry loses considerable sums of money due to operations under faulty conditions, whether via productivity losses or equipment crashes. There is also the safety problem which in some plants can be a serious concern. Under these premises, a good, fast and efficient fault detection and diagnosis (FDD) system is essential to minimize such problems. Modelling methods for FDD can be broadly classified in quantitative model-based methods, qualitative model-based methods, and process history based methods [1], [2], [3].

Over the years, methods based on processes historical data have gained great attention. One of the reasons is that these methods do not demand knowledge on the physics of the process, which can be very hard to obtain in complex systems. They rely only on acquired data during plant operation. Some examples of history based methods are neuro-fuzzy models [4], [5], neural networks [6], immune systems [7], [8], statistical methods [9], [10] and expert systems [11], [12].

Multivariate statistical process monitoring (MSPM) for fault detection has received considerable attention lately, both as a research field and in industrial applications [13]. The basis of MSPM is to find a statistical model of the normal operating condition, followed by the calculation of a monitoring statistics and its confidence bound. An abnormal condition is detected when the monitoring statistics exceed the confidence bound. For decades, the principal component analysis (PCA) for MSPM was subject of intense study and research [14], [15], [16].

The PCA based MSPM assumes a multivariate Gaussian distribution of the score, which rarely apply for a complex or non-linear process. To solve this problem, extensive research on a probabilistic framework of the PCA resulted in the probabilistic PCA (PPCA) [17], [18], [19]. Now with the statistical model of the PPCA, the generalisation to a mixture model allowed the development of more complex and complete models [21], so the MSPM became more thorough and robust.

Most of the existing offline historical data FDD methods require data from all possible operation conditions of the modelled plant, which is not a realistic scenario. Thus, the ability to learn a new behaviour incrementally is an important feature for a FDD system [1]. In this work, a batch-incremental method is proposed, which is able to learn new faulty conditions every time a data batch containing yet an unknown fault is detected. The algorithm begins training a mixture of PPCA to model only the normal operation condition of the plant, which usually has largely available data. Then data batches are analysed incrementally, looking for abnormal behaviour. Once found, the abnormal samples are used to fit another mixture of PPCA, modelling the new faulty condition. When a another batch containing the same condition is presented, the system is able to diagnose it, since it has already learnt this faulty condition.

The rest of the paper is organised as follows: Section II describes the PPCA, along with its generalisation for mixture modelling and an greedy algorithm used to find the ideal number of mixture components. Next, the proposed batch-incremental FDD method is described in the Section III. Numerical experiments results are shown in the Section IV, in which the Tennessee Eastman Process under closed loop control was considered. And finally, Section V concludes the paper and suggests further works, complementing and improving the methodology.

II. Probabilistic PCA

Principal component analysis (PCA) is a largely used model in data analysis and processing given its dimensionality reduction property. It can be summarised as the optimal projection into the principal subspace which maximizes the variance in the projected space [20]. However, the classic PCA approach does not associate a probabilistic model to the observed data. With that in mind, a PCA formulation under a probabilistic framework has been proposed based on a Gaussian latent variable model, closely related with the statistical factor analysis [17].

The PCA projects a d-dimensional observed data $\mathbf{t}$ into

a corresponding q-dimensional vector of latent variables $\mathbf{x}$, maximising the variance, such as,

$$\mathbf{t} = W\mathbf{x} + \mu + \epsilon \ , \tag{1}$$

where the $d \times q$ matrix W is composed by the q eigenvectors associated with the q highest eigenvalues of the covariance matrix, the parameter μ permits to work with non-zero mean data and ϵ is a noise. The probabilistic principal component analysis (PPCA) model assumes, basically, the latent variable to be independent and Gaussian with unit variance, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I)$, and the error noise to be isotropic, $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ [17]. Leading to a conditional probability distribution over the $\mathbf{t}$-space given by

$$p(\mathbf{t}|\mathbf{x}) \sim \mathcal{N}(W\mathbf{x} + \mu, \sigma^2 I) \ . \tag{2}$$

The marginal distribution for the observed data $\mathbf{t}$ can be derived and found likewise Gaussian:

$$p(\mathbf{t}) \sim \mathcal{N}(\mu, C) \ , \tag{3}$$

in which $C = WW^T + \sigma^2 I$ is the covariance matrix of the model. In comparison with a multivariate Gaussian model, where the sample covariance $d \times d$ matrix S needs to be estimated, the PPCA covariance matrix, C, depends only on W and σ^2. Therefore, the PPCA can be understood as a way to constrain the model complexity via the selection of $q < d$. In other words, the number of free parameters in C, $dq + 1 - q(q-1)/2$, is smaller than the $d(d+1)/2$ parameters of S, if $q < d$ [17]. All model parameters (W, μ and σ^2) can be estimated using the maximum likelihood estimation (MLE) method [17].

This probabilistic perspective for the principal component analysis brings a number of important advantages [17], but in the context of this paper two stand out among them. Firstly, it provides a single statistic for fault detection, as opposed to both the T^2 [14], which can be interpreted as a measuring of the systematic variations of the process, and the SPE [15], related to the lowest eigenvalues and associated to noise measurements [16], in the classical PCA. Since the T^2 and the Q statistics monitor different kinds of faults and use different measuring units, they can not be directly unified into only one index [18]. On the other hand, in the PPCA model, a data point $\mathbf{t}$ should be considered out-of-control (out of the given model) when

$$M^2 = (\mathbf{t} - \mu)^T C^{-1} (\mathbf{t} - \mu) > \mathcal{X}_d^2(\beta) \ , \tag{4}$$

where $\mathcal{X}_d^2(\beta)$ is the β-fractile of the chi-squared distribution with d degrees of freedom [19].

Secondly, although the PPCA is a linear model, it can be easily extended to a mixtures of such models, allowing to work with non-linear operation conditions or multiple operation points conditions.

A. Mixture Model

A PPCA mixture model with k mixture components is defined as [21]:

$$p(\mathbf{t}) = \sum_{i=1}^{k} \pi_i p(\mathbf{t}|i) \ , \tag{5}$$

where $p(\mathbf{t}|i) \sim \mathcal{N}(\mu_i, C_i)$ is the distribution of the i-th component, which $C_i = W_i W_i^T + \sigma_i^2 I$, and the mixing weights sum to the unity, $\sum_i \pi_i = 1$. Therefore, each of the k components can be understood as a local model, which combined represents the whole data set. Although there is not a closed-form expression for the parameters estimation, the MLE can be iteratively reached through the EM algorithm [21].

The motivation to use a mixture model in the FDD context is to monitor a process with multiple operating modes and/or a process in which an operating mode does not have only one single Gaussian scattering pattern in the observed variable space.

A confidence bound can be used with the PPCA mixture model to reason whether or not a data point belongs to a given model. It can be used, for example, a numerical approach such as the following Monte Carlo simulation method [22]:

1) Generate N_s samples, $\{\mathbf{t}_j \mid j = 1, ..., N_s\}$, from $p(\mathbf{t})$ given by Eq. 5.
2) Compute the likelihood of these samples as $p(\mathbf{t}_j)$.
3) Sort $p(\mathbf{t}_j)$ of all the $j = 1, ..., N_s$ samples.
4) The confidence bound is given by $h = p(\mathbf{t}_l)$, where $l = \beta N_s$.

Hence, a given data point $\mathbf{t}_i$ is considered out of control, with β-level of significance, if $p(\mathbf{t}_i) < h$, or equivalently, if $p(\mathbf{t}_i) > -h$. Typically, the number of Monte Carlo samples, N_s, has to be large, and can be determined heuristically. If the training dataset used to fit the PPCA mixture model is large enough, the confidence bound can be calculated based on those values.

B. Greedy Learning of a Mixture Model

The EM algorithm for a mixture model estimation suffers from two main drawbacks [23]. Firstly, the number of mixture components k has to be set *a priori*, requiring a knowledge of the dataset in hand, which is not always available or possible. Secondly, the parameters initialisation for the iterative estimation affects the final result. Therefore, the EM algorithm can converge to a local optimum of the parameter space.

In the PPCA mixture model, a third problem arises, that is the choice of the value q, the number of principal components, which can actually vary from one mixture component to another.

Based on a greedy learning algorithm for Gaussian mixture models [24], a greedy learning algorithm for fitting a PPCA mixture model is proposed. The basic idea is, instead of starting off with a random initialisation of the components, the model is built component-wise, using a top-down approach. That means, initially, an optimal one-component PPCA mixture is computed. Then, a new component is inserted and the

whole new mixture model is trained via EM algorithm until convergence. This process is repeated, adding new components at each iteration, until a stopping criterion is reached.

With this greedy method, the EM algorithm convergence problem is avoided by adding optimal new components, as explained later. Another advantage of the top-down approach is that since the one-component PPCA mixture model has a closed-form expression for the MLE, it does not suffer from random initialisation problems. Finally, the number q of principal components is chosen to be equal for all mixture components, and defined through the analysis of the retained variance [25].

Let f_k denote a k-component PPCA mixture model, in which each component is defined by $p(\mathbf{t}|i) \sim \mathcal{N}(\mu_i, C_i)$. The general scheme for the greedy learning of the PPCA mixture model is as follows:

1) Define q, the number of principal components, through the retained variance analysis [25].
2) Set $k = 1$ and compute f_k, the optimal one-component PPCA mixture model.
3) Create km candidate components, $p_c(\mathbf{t}|j) \sim \mathcal{N}(\mu_j, C_j)\ \forall j = 1, ..., km$ with their respective weights π_j.
4) Select the optimal new component, $p_c(\mathbf{t}|j^*) \sim \mathcal{N}(\mu_{j^*}, C_{j^*})$, and the corresponding mixing weight ,π_{j^*}, that maximizes the new log-likelihood of the whole model:

$$j^* = \arg\max_j \sum_{i=1}^{n} \log\left[(1-\pi_j) f_k(t_i) + \pi_j p_c(\mathbf{t}_i|j)\right] , \tag{6}$$

5) Set $f_{k+1} = (1-\pi_{j^*}) f_k + \pi_{j^*} p_c(\mathbf{t}|j^*)$ and $k = k+1$.
6) Update f_k with EM algorithm until convergence.
7) If stoping criterion is met (k_{max}, for example) then exit, else go to step 3.

Clearly, the candidates creation and selection is the crucial step of the previous algorithm. In [24], the author describes a search heuristic for finding the optimal new mixtures component in Gaussian mixture model. This algorithm was adapted to the PPCA mixture model scenario, and the search strategy can be summarised as follows:

1) Given a k-component mixture model f_k, a fixed number of m candidate is created for each mixture component. In other words, the number of candidates increases linearly with k, creating km candidates.
2) The data set $\mathbf{T}_n$ is partitioned in k disjoint subsets, based on the mixture components posterior distributions, such that $A_j = \{\mathbf{t}_i \in \mathbf{T}_n : j = \arg\max_j p(\mathbf{t}_i \mid j)\}$, $\forall j = 1, 2, ..., k$.
3) For each subset A_j, m candidates are created, two at a time. To generate candidate from A_j, two data points in A_j are randomly selected, $\mathbf{t}_l$ and $\mathbf{t}_r$.
4) A_j is again separated into two disjoint subsets, A_{jl} and A_{jr}, with the nearest neighbours of $\mathbf{t}_l$ and $\mathbf{t}_r$, respectively.
5) Two PPCA candidate models are finally created based on the data in A_{jl} and A_{jr}, directly with the MLE. The mixing weight of each one of the candidates is proportional to the size of its subset, $\pi_{jl} = \pi_j \frac{size(A_{jl})}{size(A_j)}$ and $\pi_{jr} = \pi_j \frac{size(A_{jr})}{size(A_j)}$, and, therefore, they sum up to the weight of the original mixture component, such that $\pi_j = \pi_{jl} + \pi_{jr}$,

The described method achieves two important observations discussed in [24]. The size of the new component, in general, should be smaller than the original mixture component, and as the model f_k grows, the search for an optimal new component should become more thorough.

This greedy approach is mainly focused on finding k, i.e., the ideal number of components in the PPCA mixture. After the algorithm runs for several values of k, it is time to reason on which is the ideal value. Through the analysis of all the mixture models created, k can be chosen, for example, via the Akaike Information Criteria, Maximum Likelihood Ratio Test, Bayesian Information Criteria, etc.

An example is given in the Figure 1, for a 2-dimension artificially generated data from a 3 component Gaussian mixture model. It shows steps from the greedy algorithm, with a mixture model with 3 and 5 components, and the log-likelihood, AIC and BIC curves of the fitted models, for $k = 1, \cdots, 10$, for analysis. It is clear that for this given data, $k = 3$ is the ideal number of components for the PPCA mixture model.

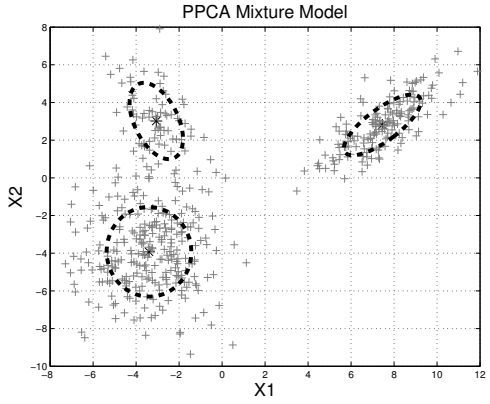
III. Proposed Method

In this section the proposed batch-incremental FDD system is described. An important feature of the proposed FDD system is that it gradually learns new faulty operation conditions of a process by analysing data batches incrementally. The proposed method begins only with the model of a normal operation condition (NOC). The greedy algorithm described in Section II-B is used to fit a PPCA mixture model that describes the normal condition. Every time a new operation condition is detected, a new mixture PPCA is estimated to model this new condition.

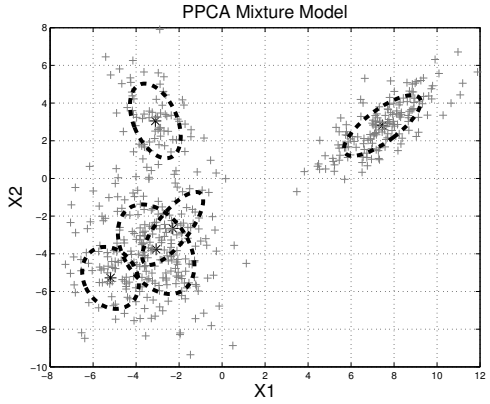
Given the NOC PPCA model, process monitoring is performed using the threshold derived in Section II-A, through the Monte Carlo simulations. By monitoring the samples' statistics of a given data stream under the normal operation model, deviations from the NOC can be detected, suggesting a fault in the process.

Suppose a new data batch to be analysed, containing N samples. Given that the threshold h of the normal operation PPCA mixture model was obtained with a $100\beta\%$ confidence bound, it is expected that, approximately, $N_f = (1-\beta)N$ samples to be detected outside the threshold, even though the process is operating under normal condition. More formally, the number of samples detected as out of control, N_f, follows a binomial distribution, $N_f \sim \mathcal{B}(N, 1-\beta)$, given that the data is under NOC. Therefore, to actually detect an abnormal operation, the number of out of control samples N_f detected must be greater than the binomial inverse cumulative distribution, also with β level of significance, lets denote $N_f > \mathcal{B}_\beta^{-1}(N, 1-\beta)$.

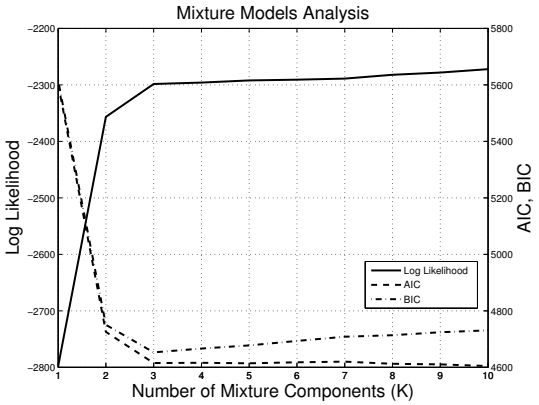
Before fitting a new PPCA mixture model for the detected fault, the N_f samples are analysed based on all the others PPCA mixture models already created (each one describing



(a) Three-components mixture.



(b) Five-components mixture.



(c) Mixture analysis.

Fig. 1. Example of some steps of the greedy algorithm.

one operation condition already observed) to check if the current data is from an already known condition. Suppose that besides the PPCA mixture model for the normal condition, another C mixture models have already been fitted to model C abnormal conditions. The idea is the same as used to check if the data deviates from the NOC. Since all C of the mixture models has its own threshold, h_c, also obtained by the Monte Carlo simulations, the same test with the binomial distribution is made, for each mixture models.

Therefore, the N_f samples are only used to fit a new PPCA mixture model if none of the already known behaviours can model them well enough. Formally, if N_{fc} is the number of samples in the subset which are outside the threshold h_c of the PPCA mixture model c, a new behaviour is detected if $N_{fc} > \mathcal{B}_\beta^{-1}(N_f, 1-\beta)$, for all $c = 1, 2, ..., C$. If $N_{fc} <= \mathcal{B}_\beta^{-1}(N_f, 1-\beta)$ for at least one c, the detected fault is already known, and the corresponding PPCA mixtures model is the one with the smallest N_{fc}.

Given that $(1 + C)$ PPCA mixture models already exists, one for the NOC and C for abnormal behaviours, such that $p(\mathbf{t} \mid c)$ for all $c = 1, 2, ..., C$, the method can be summarised as follows:

1) Given a new data batch, analyse all the N samples from the batch through the statistics $p(\mathbf{t}_i \mid \text{normal})$, detecting N_f out of control samples, under the threshold h.
2) If $N_f <= \mathcal{B}_\beta^{-1}(N, 1-\beta)$, the data batch is classified as under normal condition, and returns. Else, abnormal condition is detected and analysed in Step 3.
3) Compute $N_{fc} = \sum_{i=1}^{N_f} [p(\mathbf{t}_i \mid c) < h_c]$ for all $c = 1, 2, ..., C$, that is the number of samples in N_f that their statistics are beyond the threshold of each of the mixture models.
4) If $N_{fc} > \mathcal{B}_\beta^{-1}(N_f, 1-\beta)$, for all $c = 1, 2, ..., C$, a new behaviour is detected, go to Step 5. Else, an already known behaviour is detected, classified as $c = \arg\min_c N_{fc}$.
5) Fit a new PPCA mixture model with the greedy algorithm based on the N_f dataset, and compute its threshold h_{c+1} via Monte Carlo simulations. Add the new PPCA mixture model to the FDD system, $C = C + 1$.

These steps can also be seen in the flowchart in the Figure 3.

As previously discussed, even a normal operation condition data set will have some samples detected as abnormal, due to the β level of confidence. To minimize false alarms during these situations, an exponential moving average (EMA) filter is used in the monitoring statistics, adding some time dependency between the samples, a very reasonable assumption in the real-time analysis. The filter is given by:

$$p(\mathbf{t}_i) = (1-\gamma)p(\mathbf{t}_{i-1}) + \gamma p(\mathbf{t}_i)\ , \tag{7}$$

where γ is the filter weight. When $\gamma = 1$, the filtering action is disabled.

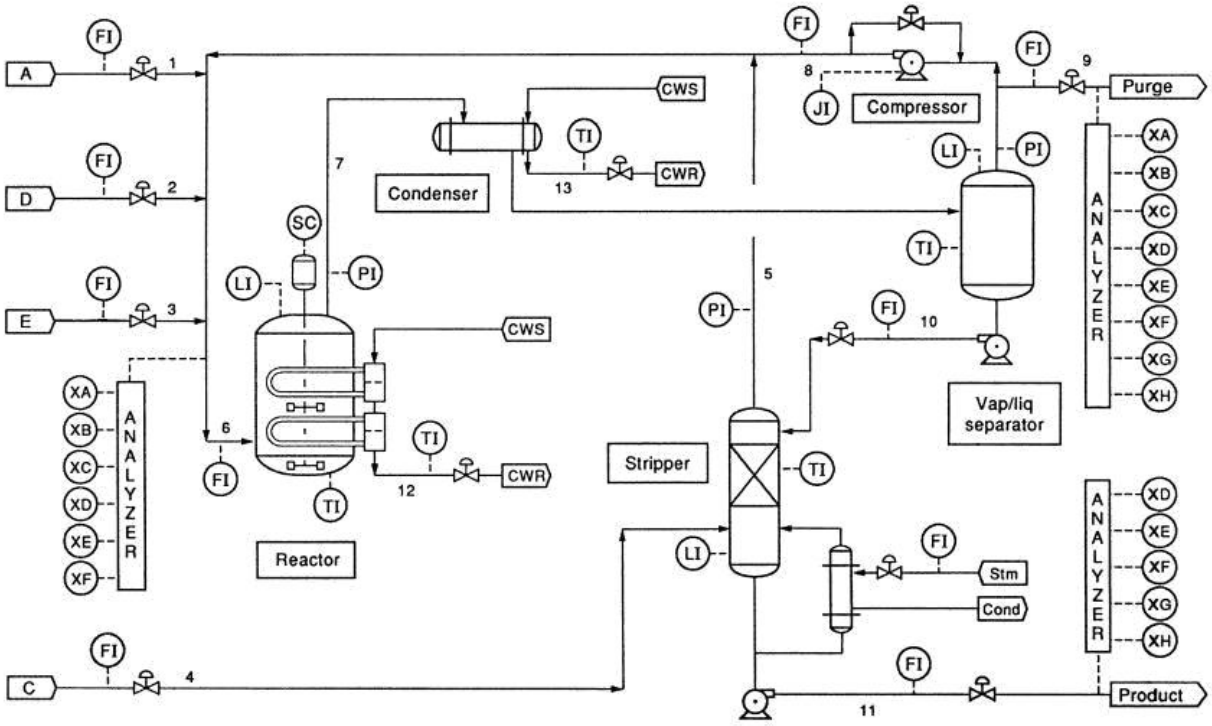


Fig. 2. Tennessee Eastman Process [27].

IV. Numerical Experiments

The proposed method was tested using a *Simulink* model of the Tennessee Eastman Process (TEP) [27] under a decentralised control strategy [28]. The process has five main units, the reactor, condenser, separator, stripper and compressor, as shown in Figure 2. The plant has eight components: A, B, C, D, E, F, G and H. The components A, C, D and E are gaseous reactants and B is a inert gas which are fed to the reactor, where the liquids G and H are produced. The component F is a subproduct of the reactions.

The process has 41 measured variables and 12 manipulated variables. Due to a uniformity in the sample period of 6 minutes, only the first 22 measured variables were used, which are summarised in the Table I. The others 19 measured variables have different sample frequency and are measurements of the plant components.

The simulation model has 21 operation modes, corresponding to one normal operation and 20 faulty operations. The majority of them were tested and successfully detected with the proposed method. The faults considered are summarised in the Table II, together with the number of samples taken to detect the fault after it occurred and the number of mixture components necessary to model its behaviour. Fault 6 presented an unstable behaviour, making it inappropriate to work with. The unlisted faults did not present enough variation in their distribution over the variables space while in the stationary state when compared to the NOC. It was detected only a quick abnormal behaviour during the transient, making it inappropriate to fit a PPCA mixture model for these faults.

In the Figure 4, it can be seen that for at least 95% of variance retainment, through the analysis of the NOC dataset, $q = 14$ principal components are necessary. As the NOC in the simulation is composed by a single process operation point, once normalised, the data scatter in the variable space is practically Gaussian, hence, an one-component PPCA mixture model was chosen to model the NOC.

TABLE I. TEP's measured variables used in the analysis.

Variable	Description
$t(1)$	Feed A (Stream 1)
$t(2)$	Feed D (Stream 2)
$t(3)$	Feed E (Stream 3)
$t(4)$	Total Feed (Stream 4)
$t(5)$	Recycle Flow (Stream 8)
$t(6)$	Reactor Feed Rate (Stream 6)
$t(7)$	Reactor Pressure
$t(8)$	Reactor Level
$t(9)$	Reactor Temperature
$t(10)$	Purge Rate (Stream 9)
$t(11)$	Product Separator Temperature
$t(12)$	Product Separator Level
$t(13)$	Product Separator Pressure
$t(14)$	Product Separator Underflow (Stream 10)
$t(15)$	Stripper Level
$t(16)$	Strippel Pressure
$t(17)$	Stripper Underflow (Stream 11)
$t(18)$	Stripper Temperature
$t(19)$	Stripper Steam Flow
$t(20)$	Compressor Work
$t(21)$	Reactor Cooling Water Outlet Temperature
$t(22)$	Separator Cooling Water Outlet Temperature

The time signal of all the 22 measured variables for a given dataset can be seen in Figure 5. In the beginning, under NOC, all variables are around zero with unit variance, since they have been normalised. The Fault 2 is applied in the sample time

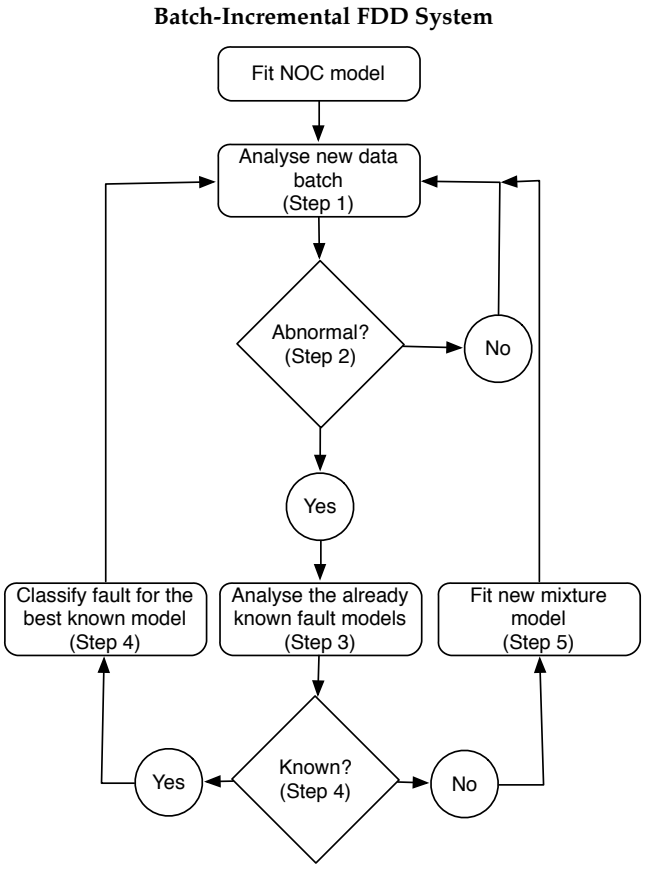


Fig. 3. Flowchart of the proposed batch-incremental FDD system.

TABLE II. PROCESS FAULTS.

Fault ID	Description	Type	Detection Delay	No of Mixture Components
1	A/C feed ratio, B composition constant (Stream 4)	Step	6	5
2	B composition, A/C ratio constant (Strem 4)	Step	6	2
8	A, B, C feed composition (Stream 4)	Random variation	11	4
10	C feed temperature	Random variation	22	1
11	Reactor cooling water inlet temperature	Random variation	6	1
12	Condenser cooling water inlet temperature	Random variation	13	1
13	Reaction kinetics	Slow drift	13	4
14	Rector cooling water valve	Sticking	6	1
17	Unknown	Unknown	21	5
18	Unknown	Unknown	41	2
19	Unknown	Unknown	35	1
20	Unknown	Unknown	28	2

100, causing a transitional behaviour at first, stabilising after a while. Running this dataset through the proposed method, the fault was correctly detected. In the Figure 6 the monitoring statistic (already filtered, with $\gamma = 0.9$) is shown for each sample, together with the threshold h for the normal operation condition, where can be seen that the threshold was exceeded 6 samples after the fault occurred.

Since this fault is yet unknown, all the samples in which its monitoring statistics is classified as abnormal are used to

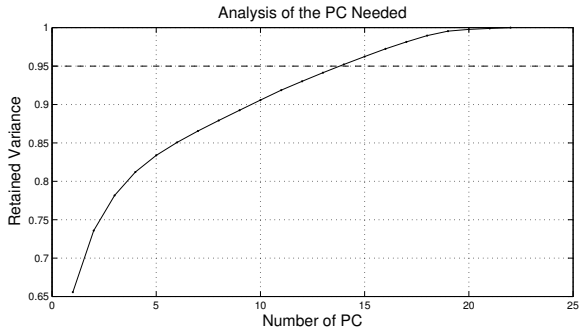


Fig. 4. Analysis the the retained variance for the choice of q, the number of principal components.

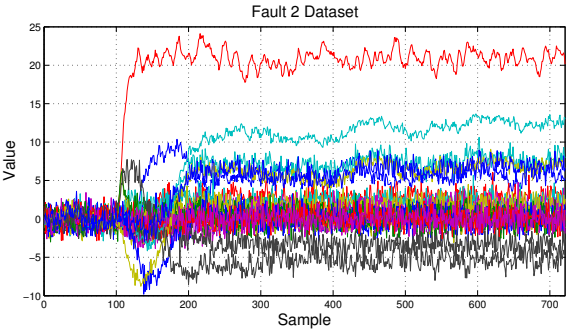


Fig. 5. Dataset for the Fault 2.

train a new PPCA mixture model, so if this fault happens once again, the FDD system should be able to diagnose it. The trained model for the Fault 2 has a 2-component PPCA mixture model, chosen from the analysis shown in the Figure 7, with the same $q = 14$ principal components.

Supposing a situation, where a new data batch to be analysed contains the same fault, the first analysis would show again an abnormal behaviour, similar to the Figure 6, but now analysing the abnormal samples with the already trained PPCA mixture model for this fault, as shown in the Figure 8, the analysed behaviour can be classified correctly as Fault 2, whereas the number of samples beyond the model threshold is less then the expected binomial distribution.

For a more complete simulation, a sequence of operation conditions, divided by data batches fed separately, was analysed by the proposed algorithm. Such sequence follows as described in Figure 9, where the faults 2, 11, 13 14 and 20 were chosen so it covered all fives types of faults in the process. Each data batch begins from the NOC, represented as the Operation ID 0, and since normally the plant does not jump from one fault condition to another, it returns to the NOC after the fault. Also, the chosen faults were applied twice in order to verify the diagnostic part of the method. All the 22 measured variables for this sequence, just for a rough visualisation of the data, can be seen in the Figure 10.

The results are shown in the Figure 11. The dotted line shows the same sequence of operations as in the Figure 9, for a matter of reference. The asterisk symbol shows the time instant where the given fault was detected and the black diamond symbol shows the samples used to fit a new model, when an

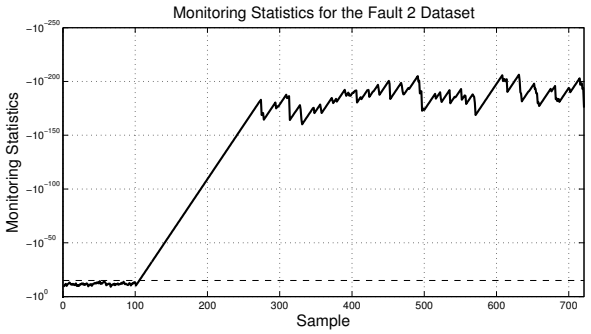


Fig. 6. Monitoring statistics for the Fault 2 dataset, in logarithm scale.

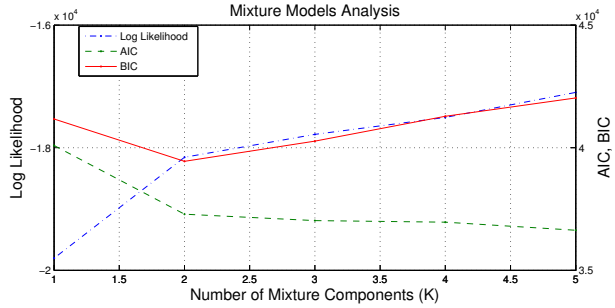


Fig. 7. Analysis to choose the number o mixture components for the Fault 2 PPCA mixture model.

unknown behaviour is detected. The dashed line is the classifier output.

It can be seen that on the first instance of the given faults, the faulty samples are used to fit a PPCA mixture model to describe such condition and, as it is yet unknown, there is no output from the classifier. When an already known fault occurs, the classifier is able to diagnose the given fault, as seen in the last five faults of the sequence.

Furthermore, if seen closely in Figure 10, the faults 2 and 13 have a slightly longer transient. Since a PPCA mixture model was used, these samples were also successfully modelled, as shown in Figure 11, both in the abnormal samples chosen to fit the model and the classifier output, where it extrapolates the "step" of the fault. In conclusion, the proposed method correctly detects a new behaviour and successfully diagnose a fault once it is already known.

V. Conclusions

This paper proposes a batch-incremental method for process fault detection and diagnosis (FDD). The historic data based modelling uses mixtures of probabilistic principal component analysis (PPCA) as the base to model the operation conditions of the plant. The PPCA models are estimated via a greedy algorithm that helps identify the ideal number of mixture components necessary to model a given behaviour and avoid convergence problems.

The use of a mixture of PPCA allows to model operation conditions with non-gaussian data patterns in high dimensional variable spaces, which broadens the possibilities of application. The proposed method has shown promising results when applied to the Tennessee Eastman Process problem, being able

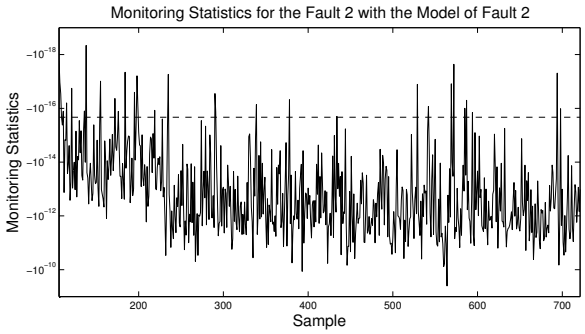


Fig. 8. Monitoring statistics for the Fault 2 dataset, in logarithm scale, for the PPCA mixture model of the Fault 2.

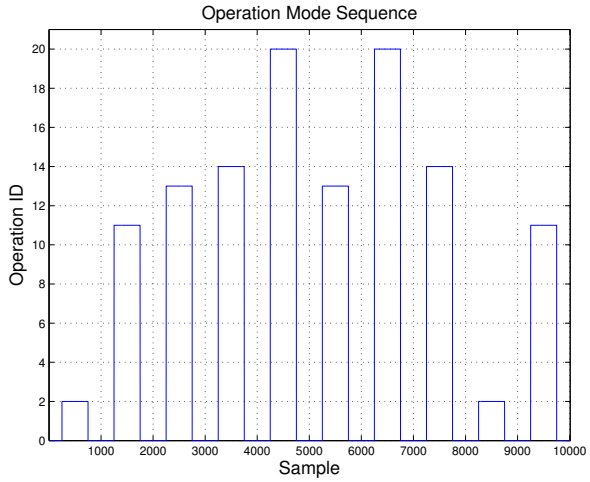


Fig. 9. Sequence of operations conditions applied to the proposed method.

to precisely identify faults and diagnose already known faulty conditions. Although the proposed FDD system do not have an online learning, the monitoring of a process can be done in real time starting only with a NOC data set, which can be of significant help to operators and technicians work on the process.

Future work is focused on extending the proposed method in two ways. Firstly, a probabilistic contribution analysis [29] can be used to isolate the variables associated to a given fault, directing the operators to the possible physical location of the problem, facilitating the diagnostic. Furthermore, a online adaptive learning method for the mixtures of PPCA would increase the automation level of the system, being able to adapt to changes in the general behaviour of the process.

Acknowledgement

The authors acknowledge the support of FAPEMIG, the Research Foundation of the State of Minas Gerais.

References

[1] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N.: A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-based Methods, Computers & Chemical Engineering, Volume 27, Issue 3, 293–311 (2003)

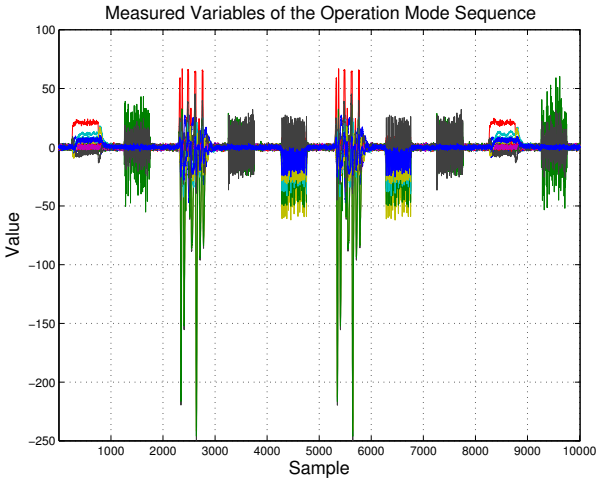


Fig. 10. Measured variables in the sequence of operations conditions applied to the proposed method.

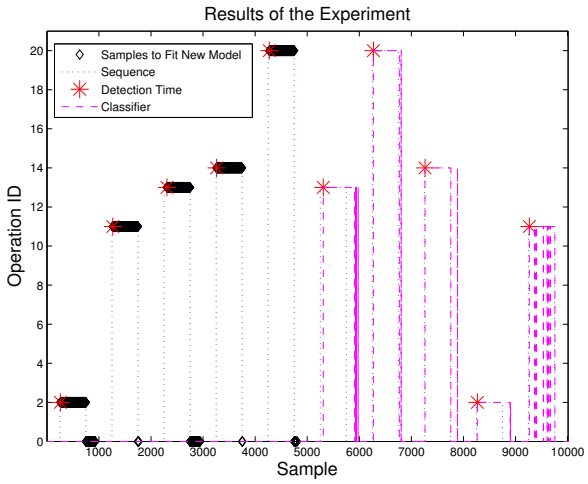


Fig. 11. Results for the sequence of operation conditions.

[2] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N.: A Review of Process Fault Detection and Diagnosis: Part II: Qualitative Models and Search Strategies, Computers & Chemical Engineering, Volume 27, Issue 3, 313–326 (2003)

[3] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N.: A Review of Process Fault Detection and Diagnosis: Part III: Process History Based Methods, Computers & Chemical Engineering, Volume 27, Issue 3, 327–346 (2003)

[4] Lemos, A., Caminhas, W., Gomide, F.: Adaptive Fault Detection and Diagnosis Using an Evolving Fuzzy Classifier, Information Sciences, Volume 220, 64–85 (2013)

[5] Mendonca, L.F., Sousa, J.M.C., da Costa, J.M.G.S.: An Architecture for Fault Detection and Isolation Based on Fuzzy Methods, Expert Systems with Applications 36, 1092–1104 (2009)

[6] Samanta, B.: Gear Fault Detection Using Artificial Neural Networks and Support Vector Machines with Genetic Algorithms, Mechanical Systems and Signal Processing 18, 625–644 (2004)

[7] Laurentys, C.A., Palhares, R.M., Caminhas, W.M.: A Novel Artificial Immune System for Fault Behavior Detection, Expert Systems with Applications 38, 6957–6966 (2011)

[8] Laurentys, C.A., Palhares, R.M., Caminhas, W.M.: Design of an Artificial Immune System Based on Danger Model for Fault Detection, Expert Systems with Applications 37, 5145–5152 (2010)

[9] Mehranbod, N., Soroush, M., Panjapornpon, C.: A Method of Densor Fault Detection and Identification, Journal of Process Control, 15 321339 (2005)

[10] D'Angelo, M.F., Palhares, R.M., Takahashi, R.H., Loschi, R.H., Baccarini, L.M., Caminhas, W.M.: Incipient Fault Detection in Induction Machine Stator-Winding Using a Fuzzy-Bayesian Change Point Detection Approach, Applied Soft Computing 11, 179–192 (2011)

[11] Angeli, C., Chatzinikolaou, A.: On-line Fault Detection Techniques for Technical Systems: A Survey, International Journal of Computer Science & Applications 1, 12–30 (2004)

[12] Cho, H.J., Park, J.K.: An Expert System for Fault Section Diagnosis of Power Systems Using Fuzzy Relations, IEEE Transactions on Power Systems 12, 342–347 (1997)

[13] Joe Qin, S.: Statistical process monitoring: basics and beyond, Journal of Chemometrics, Vol. 17, 480–502 (2003)

[14] Hotelling, H.: The Generalization of Student's Ratio, The Annals of Mathematical Statistics, Institute of Mathematical Statistics, Vol. 2, N. 3, 360–378 (1931)

[15] Jackson, J.E., Mudholkar, G.S.: Control Procedures for Residuals Associated with Principal Component Analysis. Technometrics, American Statistical Association and American Society for Quality, Vol. 21, N. 3, 341–349 (1979).

[16] Villegas, T., Fuente, M. J., Rodríguez, M.: Principal Component Analysis for Fault Detection and Diagnosis. Experience with a Pilot Plant, Proceedings of the 9th WSEAS International Conference on Computational Intelligence, Man-machine Systems and Cybernetics, 147–152 (2010)

[17] Tipping, M. E., Bishop, C. M.: Probabilistic Principal Component Analysis, Journal of the Royal Statistical Society, Series B, Vol. 61, 611–622 (1999)

[18] Kim, D., Lee, I.-B.: Process monitoring based on probabilistic PCA, Chemometrics and Intelligent Laboratory Systems, Vol. 67, N. 2, 109–123 (2003)

[19] Chen, T., Sun, Y.: Probabilistic contribution analysis for statistical process monitoring: A missing variable approach, Control Engineering Practice, 17 (4). 469–477 (2009)

[20] Hotelling, H.: Analysis of a Complex of Statistical Variables Into Principal Components, Journal of Educational Psychology, Vol. 24, 417-441 and 498-520 (1933)

[21] Tipping, M. E., Bishop, C. M.: Mixture of Probabilistic Principal Component Analysers (1998)

[22] Chen, T., Morris, J., Martin, E.: Probability Density Estimation via an Infinite Gaussian Mixture Model: Application to Statistical Process Monitoring, J ROY SOC C-APP, 55, 699-715 (2006)

[23] Ververidis, D., Kotropoulos, C.: Gaussian Mixture Modeling by Exploiting the Mahalanobis Distance, Signal Processing, IEEE Transactions, Vol. 56, 2797–2811 (2008)

[24] Verbeek, J.J., Vlassis, N., Kröse, B.: Efficient Greedy Learning of Gaussian Mixture Models, Neural Computation, Vol. 15, 469–485 (2003)

[25] Jolliffe, I.: Principal Component Analysis, Wiley Online Library (2005)

[26] Gama, J., Medas, P., Castillo, G., Rodrigues., P.: Learning with drift detection, IN SBIA Brazilian Symposium on Artificial Intelligence, 286–295 (2004)

[27] Downs, J. J., Vogel, E. F.: A plant-wide industrial process control problem. Computers & Chemical Engineering, Vol. 17 (3), 245–255 (1993)

[28] Ricker, N. L.: Decentralized control of the Tennessee Eastman Challenge Process", Journal of Process Control, Vol. 6, No. 4, 205–221 (1996)

[29] Kariwalaa, V., Odioweib, P. E., Caob, Y., Chen, T.: A branch and bound method for isolation of faulty variables through missing variable analysis, Journal of Process Control, Vol. 20, N. 10, 11981206 (2010)