A SELF-ORGANIZING MAP APPROACH FOR PROCESS FAULT DIAGNOSIS DURING PROCESS TRANSITIONS

Y.S. Ng and R. Srinivasan*

Laboratory for Intelligent Applications in Chemical Engineering,
Department of Chemical and Environmental Engineering,
National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260
**Corresponding Author: Email: chergs@nus.edu.sg*

Abstract: In this paper, we outline a self-organizing map (SOM) based approach to monitor process transitions. The framework integrates SOM with clustering and sequence comparison methods for plant wide monitoring and fault diagnosis. Process abnormality is detected through cluster analysis while syntactic pattern recognition technique and profile sequence comparison techniques render data based fault diagnosis and machine learning possible. Furthermore, the proposed method also inherits the powerful visualization facility of SOM. Extensive testing on the operations of a lab-scale distillation column illustrates the method's efficacy.

Keywords: monitoring, fault diagnosis, self-organizing map, cluster analysis.

1. INTRODUCTION

Transitions are quite common in the process industries. Transitions occur during startup, shutdown, feedstock changes, product slate changes, etc in chemical processes. Transitions commonly entail large changes in the plant operating conditions, thus hindering the effective operations of current day control systems. Similarly, most of the high-level automation applications are effective only during steady states. Plant operators therefore perform transitions manually following predefined standard operating procedures (SOP), which clearly state the sequence of actions that need to be taken, e.g. open or close valves, activate or deactivate equipments, reconfigure controllers, etc. Owing to the lack of effective automation and the high cognitive workload for operators, the occurrence of human errors during transitions is more common. Surveys in the US, Canada, UK, Europe, and Japan reveal that human errors, especially during transitions, are the leading cause of abnormal situations (Nimmo, 1995). With the growing scale of chemical plants and the complexity and agility of process operations necessitated by market conditions, monitoring of transitions remains a challenging problem. In this paper, we outline a self-organizing map (SOM) based approach to perform plant-wide monitoring and fault diagnosis. The method inherits the powerful visualization facility of SOM and enables multivariate monitoring on a two-dimensional map. The integration with clustering and sequence alignment technique also render automated data-based process monitoring and fault diagnosis possible.

The organization of this paper is as follows: Section 2 presents an introduction to SOM and some of its applications in the process industries. Section 3 describes the SOM methodology for monitoring transitions while Section 4 presents a case study from the startup of a distillation column.

2. THE SELF-ORGANIZING MAP

The self organizing map was first proposed by Kohonen in 1981 as a visualization tool, but has since become one of the most popular neural network architectures. SOM belongs to the unsupervised learning type of neural networks and is capable of projecting high-dimensional input onto a lower, usually two-dimensional grid. SOM employs nonparametric regression and involves the fitting of discrete, ordered reference vectors to the distribution of input feature vectors. A finite number of reference vectors are adaptively placed in the input signal space to approximate input signals. Self-organization means that the net orients and adaptively assumes a form by which it best describes the input vectors in an ordered, structured fashion (Kohonen, 1993). Consider a n-dimensional input vector, $\bar{x}$, given by

$$\bar{x} = \{x_1, x_2, x_3, ..., x_n\} \qquad (1)$$

and m_i a parametric real vector in the same space

$$\overline{m}_i = \{m_1, m_2, m_3, ..., m_n\} \qquad (2)$$

Each $\overline{m}_i$ also represents a node on a output grid (usually hexagonal). The map unit $\overline{m}_i$ which gives the smallest Euclidean distance with $\bar{x}$ is defined as the best-matching unit (*BMU*), represented here as

$$\overline{m}^{BMU} = \arg\min_i (\| \bar{x} - \overline{m}_i \|) \qquad (3)$$

During the training of the SOM, the reference vector of the *BMU*, $\overline{m}_i$ as well as those of its topological neighbors is updated by moving it towards the training sample $\bar{x}$. The SOM learning rule at iteration *t* is given by

$$\overline{m}_i(t+1) = \overline{m}_i(t) + \alpha(t) h_i^{BMU}(t)[\bar{x}(t) - \overline{m}_i(t)] \qquad (4)$$

where $h_i^{BMU}(t)$ is the Gaussian neighborhood function given by

$$h_i^{BMU}(t) = \alpha(t).\exp\left(-\frac{\| r_{BMU} - r_i \|^2}{2\sigma^2(t)}\right) \qquad (5)$$

and $\alpha(t)$ is the learning rate factor and *r* is the radius of the neighborhood function. It is necessary that as training proceeds h_i^{BMU} reduces to zero to guarantee convergence. A large value of α is employed initially and is usually decreased monotonically with *t* (Kohonen, 2000). During the training phase, SOM will fold onto the pattern formed by the training data and neighboring units are pulled nearer together because of the neighborhood relation. Thus neighboring units are, in a sense, more similar to each other. The visualization of SOM is usually done through a unified distance matrix (U-matrix) (Ultsch, 1990), where the distances of each map unit to its neighbors are calculated and displayed through gray or color-scales. An example of a U-matrix is given in Figure 1. Borders of dark color in the U-matrix can be interpreted as regions of high distances separating clusters of low distances, indicated by map units of lighter color. Interested readers are referred to Kohonen (2000) for a more detailed description of SOM.

SOM implements an ordered dimensionality reduction through the mapping of the input feature vectors while preserving the most crucial topological and metric relationships of the original data, by producing a similarity map of the input feature vectors. Some of the previous applications of SOM are presented next. Deventer *et al*. (1996) demonstrated how disturbances in a froth flotation plant can be visualized with SOM. They track changes in operating conditions through an on-line computer system that utilizes features extracted from froth images and visualize the degree of dispersion of the various input feature vectors through SOM. Chan *et al*. (2001) presented a constrained Kohonen networks to overcome the problem of monitoring redundant sensors by constraining the weight vectors in the parity space. Srinivasan and Gopal (2002) showed how SOM can be used to extract operating information from operating data of a fluidized catalytic cracking unit. Jämsä-Jounela *et al*. (2003) presented a SOM based fault diagnosis system for a smelter based on heuristic rules. SOM was used to determine the coefficient for oxygen enrichment and detection of aggregations in various parts of the plant. Abonyi *et al*. (2003) applied SOM to a polyethylene process for product quality estimation. They developed multiple local linear models for the process through piecewise linear regression with SOM.

3. PROPOSED METHODOLOGY

Process transitions commonly entail large changes in the plant operating conditions. Also, the state of the process dynamically evolves with time. This is difficult to track as the process can encompass a wide region in the self-organizing map. In our approach, process monitoring is performed by observing the time series trajectory of process operations on the SOM. The trajectory produced is time independent as it depends only on the states of the process. Process dwell time during transitions caused by run-to-run deviations is thus taken care of implicitly. Occurrence of a process fault can be observed through deviations from normal operating trajectory and root causes are identified by analyzing the component planes (Ng and Srinivasan, 2004a). However, this approach involves intensive human involvement in interpreting the process trajectory and diagnosing root cause. As an alternative, a fault database can be used to automate fault diagnosis based on syntactic pattern recognition. Monitoring statistics can also be developed to provide variable-wise residuals during an abnormal event.

Two stages of development can be differentiated. The first stage involves the construction of the SOM reference model and creation of a fault database based on the abstraction of process data into state sequence, as shown in Figure 1. This stage comprises a series of training and model updating processes, and the decision on the tuning parameters will also be decided. As can be seen in Figure 1, historical data from both normal and abnormal regions are first extracted from plant historians and projected on SOM. Next, the constructed SOM is clustered through clustering algorithm to identify regions of high similarity. Reference dataset, akin to the golden run of batch processes, are then SOM projected to update the reference model. The constructed model will later be used to monitor for discrepancies in online process measurements. The hits observed should be consistent with the SOM model when the process is normal. Any abnormal event in plant will cause abnormal hit-cluster sequence development which is inconsistent with the reference model. In this work, faults are isolated by fault database search and monitoring statistics. The steps taken to build the SOM model and fault database is presented in Section 3.1, while Section 3.2 describes the steps for online deployment.

3.1. DEVELOPMENT OF ESSOM AND FAULT DATABASE

The proposed off-line training algorithm is summarized in Figure 2. The historical dataset are first denoised and normalized before being projected to SOM. The fully trained SOM is then clustered through *k*-means clustering into a predefined set of clusters. The reference model - ESSOM (Enhanced Structure Self-Organizing Map), is developed after the clustering of SOM. ESSOM is composed of groups of cluster objects, which store the characteristics of normal operations eg: predecessor and successor of a cluster, process dwell time, etc. The training of ESSOM is done by recursive projection of different sets of reference data, or normal templates to SOM. The fully trained ESSOM, is then used as a model to detect process abnormality through state-cluster mismatch during online process monitoring. A fault database is also constructed based on the clustering results to isolate known process faults based on syntactic pattern recognition. The details of the construction of ESSOM and the fault database are elucidated next:

The raw sensor measurements are first de-noised through a finite impulse response (FIR) filter (DSP committee, 1979) to filter out high frequency noise before each variable is variance normalized. The normalized dataset, $\bar{x}$, is then projected to a SOM. Projecting the training data onto the SOM may provide a means to visualize the process operating trajectory. Post-projection analysis is essential to automate monitoring. Towards that end, the SOM is further clustered into predefined number of clusters to identify regions of high similarity. We use *k*-means (Sneath and Sokal, 1973) to partition SOM based on reference vectors of the SOM units. The number of partitions, K will affect the sensitivity and selectivity of the system in detecting abnormal events and isolating the root cause, as higher number of cluster defines more regions on SOM to better represent the characteristic of the underlying process. The user can adjust K to the desired level of sensitiveness. While too large a K will subject the system to more fault positive events, too few clusters will subject the system to more fault negatives. The *k*-means is executed for multiple replicates and the run which yields the lowest total square-error, ε, is selected. It is important that the number of replicates for *k*-means is high so as an optimum partition is obtained. The clustered regions can be used to track process trajectories across SOM. Abnormal situations are detected through cluster analysis by analyzing the hits clusters evolution as compared to a normal reference template.

The construction of ESSOM is based on the SOM clusters formed from the *k*-means clustering. A cluster object, **c** that is the primary entity of ESSOM is created for each cluster on SOM to identify features of a normal run. A total number of K cluster objects are initialized, with eight attributes each, namely, *name, centroid, activation status, incoming cluster, incoming time, outgoing cluster, outgoing time* and *hits distribution*. Sets of reference datasets, $\bar{r}_R$, akin to the golden runs of batch processes, are recursively projected to SOM to update ESSOM. The constructed ESSOM is general enough to be used for both transient and steady–state monitoring. The construction of a fault database for diagnosing fault is described next. The self-organizing map, after being clustered into K states by *k*-means, provides a means for syntactic pattern recognition as structural information can be extracted as state-sequence for further assessment. The feature representation of the syntactic pattern recognition technique is based on the hierarchical abstraction of the process features from the self-organizing map, shown graphically in Figure 1. Data from faulty operation are first decomposed hierarchically into simpler patterns, with the hits on SOM forming the primitive language to represent basic changes in the response of the dataset. The extracted hits of process trajectory are decomposed to state sequences, and the state sequences are further refined into fault signatures which will be used online for fault identification. The fault signatures created form the primary entity of a fault morpheme, defined here as a fault information spreadsheet that stores the information of a fault in the fault database. A fault morpheme is generated with its attributes reflecting the characteristics of the fault it represents. A fault morpheme has several attributes, eg: *fault-signature, dwell-time,* and *recovery actions.* If an abnormal situation is observed during operation, the online signature will be matched with the signatures of the fault morphemes in the fault database through similarity search. If successful matches are observed, the fault morphemes together with their rectification strategies will be extracted and presented to plant personnel for implementation.

3.2. ON-LINE MONITORING AND FAULT DIAGNOSIS

The ESSOM and fault database developed in the previous phase can be deployed for online fault detection and diagnosis. The process of detecting faults is based on monitoring the discrepancy between online state-sequence developments with the reference model ESSOM. Fault diagnosis algorithms are triggered upon detection of an abnormal event. Two diagnostic algorithms have been implemented in this work to help plant personnel to identify the root cause of a fault: a fault database search methodology based on syntactic pattern recognition, and variable-wise residuals monitoring based on profile comparison. One significant issue in transition diagnosis is the alignment of temporal patterns since they can be of different lengths due to operating variations. Therefore, fault signatures and process profiles have to be synchronized before they are compared. We synchronize temporal patterns using sequence alignment.

The algorithm for on-line fault diagnosis is shown in Figure 3. During online monitoring, each process variable, designated here as $\bar{y}_n$, is first normalized using the same linear transformation that was used during the offline training phase to $\bar{x}_n$, before being projected to the trained SOM. The BMU of $\bar{x}_n$, $\bar{m}_j$ is then identified and its corresponding state identified. The novelty index and the cluster sequence are constantly monitored for deviation. A novel fault is said to have occurred if the novelty index, $\eta = \sum_{n=1}^{N} |\bar{x}_{jn} - \bar{m}_n^{BMU}|^2$ crosses a predetermined threshold. The novelty index is a direct measure of the quantization error between the weight vectors of the BMU and the corresponding sensors measurement. Abnormal trajectories are detected based on cluster analysis. Whenever there is a change in state-cluster, *S*, in the monitored trajectory from *S(t-1)* to *S(t)*, where *S(t) ≠ S(t-1)*, sequence consistency is checked with the reference model, $ESSOM_{c(t)}$. If there is a violation in the cluster progression, an alarm will be flagged. The diagnostic algorithms are also triggered to provide necessary guidance to operators.

Two diagnostic methods have been implemented in this work to isolate abnormal events, namely: fault database syntactic pattern recognition, and profile comparison to quantify process variations. The syntactic pattern recognition module contains procedures to perform database search and retrieve solutions. There have been numerous approaches proposed to classify patterns based on syntactic pattern recognition. Throughout this work, a parser based syntactic pattern recognition approach has been adopted to perform fault classification (Schalkoff, 1992). The parsing of signatures syntax is automated with the sequence alignment approach. When an abnormal situation is detected, the vector storing the cluster progression, termed as the online signature of the process, is sent for pattern matching with the fault morphemes in the fault database. Three indexes are defined for the purpose of retrieving entries from the database, namely similarity degree, fault maturity degree, and specificity degree.

Similarity degree, Π, is given by $\Pi = \dfrac{sum\ of\ matched\ signature\ entites}{length\ of\ online\ signature} x100\%$, while

Specificity degree, Y, is given by $Y = \sum_{l=1}^{L} \dfrac{(online\ hits\ \text{-}\ frequency)_l - (reference\ hits\ \text{-}\ frequency)_l}{reference\ hits\ \text{-}\ frequency_l}$, and

Fault maturity degree, Γ, is given by $\Gamma = \dfrac{sum\ of\ matched\ signature\ entities}{length\ of\ fault\ morpheme} \times 100\%$.

The similarity degree and specificity degree provide a means to extract solutions from a fault database. The similarity degree measures the similarity between the observed fault and the entries in database. The search is said to be successful if it retrieves only one result and is being able to identify the root cause accurately. On the other hand, the fault maturity degree tells us how mature the fault is in the current system. Fault maturity degree tends to increase over time as the characteristics, or behavior of a fault has become more apparent. There are two desired characteristics of a sequence alignment parser. First, the sequences are compared in an optimal manner as sequence alignment produced optimally aligned sequences, which improves the proposed methods' accuracy over conventional parser, which are often difficult to built and hard to update for large system. Secondly, sequence alignment allows automatic correction of primitives. Noise or outliers of process which produces erroneous primitives are inherently taken care of during the alignment process by the introduction of gaps to isolate the erroneous primitives.

Monitoring statistic has also been formulated to generate variable-wise residuals upon detection of abnormal event. A statistical scoring scheme, D-statistic, Θ, so called because of its Dynamic nature, is defined here to help plant personnel to verify the database search results and handle novel faults. The D-statistic serves as a monitoring chart for plant personnel to monitor process variables, and produce variable-wise residuals during an abnormal event. The residuals generated are important for two reasons. First, it gives a clear overview on the deviations of process variables and helps plant personnel in deducing the root cause during an abnormal situation; and secondly, it supplements causal model such as signed diagraphs and render their usability during process transitions. The optimal reference operating condition, $\overline{m}^{opt}$ as compared to the current state of process, $\overline{x}_j$ is used to compute D-statistic, $\Theta = \frac{1}{N}\sum_{n=1}^{N} | \overline{x}_{jn} - \overline{m}_n^{opt} |$. Θ gives a direct measurement of the severity of the underlying abnormal event. A continuous increment in Θ indicates that the fault is getting more severe and immediate attention is required and similarly vice versa. While Θ is an average total error measurements between $\overline{m}^{opt}$ and $\overline{x}_j$, the variables residuals is of more important to the plant operators, since it contains process variable-wise information for causal analysis. The variables residuals will be displayed through percentage deviation, $\overline{\delta}$, defined as $\overline{\delta}_n = \frac{\overline{x}_{jn} - \overline{m}_n^{opt}}{\overline{m}_n^{opt}} \times 100\%$.

4. A DISTILLATION COLUMN CASE STUDY

The proposed framework is tested on a lab-scale distillation unit as shown in Figure 4. The distillation column is of 2 meters height and 20cm width and has 10 trays, where the feed enters at tray 4. The system is well integrated with a control console and data acquisition system. 19 variables comprising of all tray temperatures, reboiler and condenser temperature, reflux ratio, top and bottom column temperatures, feed pump power, reboiler heat duty, and cooling water inlet and outlet temperatures, are measured at 10-second intervals. Cold startup of the distillation column with ethanol-water 30% v/v mixture is performed following the standard operating procedure (SOP), shown in Table 1. The feed passes through a heat exchanger before being fed to the column. The startup normally takes two hours and different faults such as sensor fault, failure to open pump, too high a reflux ratio etc., can be introduced at different states of operation.

Experiments are first carried out separately to populate the plant historian. The faulty dataset, together with the normal reference template, are then used to train the SOM. The trained SOM for a normal startup of the unit, as shown in Figure 5, consists of 30 x 16 map units. The trained SOM was further clustered using *k*-means (*K*= 40). The *k*-means was executed for 2000 replicates, by using squared Euclidean distance for total square error computation. The ESSOM created then contains 40 cluster objects; their attributes were further updated through projection of a normal template. As can be seen in Figure 5, the startup of the unit can be easily visualized from the trained SOM. The startup process has been observed to follow a trajectory on the U-matrix by evolving from one cluster to another, these series of cluster evolutions stored the characteristics or signatures of a run. The startup process begins at cluster 36, evolves through a series of intermediate clusters before settling at cluster 21 when steady state is attained at t=3890s. Similarly, a fault database was also constructed from the faulty datasets in the plant historian for diagnosing the root cause of a fault.

Scenario 1: DST01- Reboiler power fault

Online data was projected onto the SOM and the cluster hits identified. Figure 6 shows the trajectory during fault DST01. The dark solid line corresponds to the faulty trajectory and the light solid line is the reference trajectory from the normal run. The process signals for DST01 are shown in Figure 7, where the solid lines represent the signals for the faulty operation while the dotted lines are process signals of the reference template. The process fault was introduced at 20s, resulting in long heating time and unsuccessful startup when the feed pump was activated at step 7 of SOP. The problem was successfully detected by ESSOM at time 100s when the cluster deviated from cluster 36 to 23, with plant operator being informed before the fault upset the whole startup process. The D-statistic contribution charts at time of fault detection are shown in Figure 8. From Figure 8, one can easily recognize that the reboiler power is the root cause of the fault; the proposed method thus enables early corrective actions to be taken to alleviate the abnormal event. Failure to rectify the above problem would result in further deviation from the normal operating conditions. Direct signals comparison or signals interpolation would generate erroneous results eg, direct generation of residuals at Tray 4 temperature at time 2800s would suggest a high residual for process variable T4 since they span through different state of the process; the reference template is in the boiling phase while the process is still in the reboiler heating phase. The D-statistic formulated is thus capable of producing optimal results by locating the exact location in the reference template and providing accurate variables residuals to help plant personnel in diagnosing the root cause. The characteristic of the faulty run, or its signature at time of fault detection, was also sent for database matching through syntactic pattern recognition. Plant personnel can then verify the suggested fault rectification strategy extracted by the database search algorithm by confirming it with the D-statistic contribution charts before they are implemented in the process. Failure in rectifying this fault will result in unsuccessful column startup in process plant and introduce unnecessary delay to both upstream and downstream processing units.

5. CONCLUSIONS

This paper presents the methodology in developing SOM for process fault diagnosis by incorporating clustering and sequence alignment technique. A novel syntactic pattern recognition based methodology has been proposed for classifying known process faults based on database search. The pattern recognition technique endows the proposed method with learning property to classify new process faults when such dataset become available. A statistical monitoring scheme based on sequence alignment technique has also been introduced to monitor the severity of process fault, and to generate variables residuals during abnormal events to facilitate plant wide fault diagnosis. The application of the above methods to a distillation column startup shows the method effectiveness in detecting and classifying process faults. The proposed method offers several advantages over the other monitoring techniques. It accounts the multivariate nature of chemical processes and is able to visualize high dimensional data, making it superior to most of the currently available monitoring techniques. The proposed technique is much faster than conventional signals comparison methods. In addition, the sequence comparison method is also less computational demanding, and has found to be able to supplement external causal models, e.g., signed diagraphs or observers, through the D-statistic contribution charts generated. The method is also relatively easy to scale up and can be applied to multiple platforms with minor changes in algorithms. Our group hopes that the above developed method can be a good supplement to the currently available monitoring techniques. Future work is oriented towards integrating SOM model with heterogeneous FDI models for collaborative decision support during process operations (Ng and Srinivasan, 2004b).

REFERENCES

Abonyi, J., Nemeth, S., Vincze., Arva, P., (2003). Process analysis and product quality estimation by self-organizing map with an application to polyethylene production. *Computers in Industry 52*. 221-234.

Chan, C.W., Jin, H., Cheung, K.C., Zhang, H.Y., (2001). Fault detection of systems with redundant sensors using constrained Kohonen networks. *Auomatica 37*. 1671-1676.

Deventer, J.S.J.V., Moolman, D.W., Aldrich, C., (1996). Visualization of plant disturbances using self-organizing maps, *Computers & Chemical Engineering Vol.20*, pp.S1095-S1100.

DSP Committee (eds), (1979). Programs for digital signal processing, IEEE Press, New York.

Jämsä-Jounela, S.L., Vermasvuori, M., Endén, P., Haavisto,S., (2003). A process monitoring system based on the Kohonen self-organizing maps. *Control Engineering Practice 11*. 83-92.

Kohonen, T., (1993). Things you haven't heard about the self-organizing map, *IEEE International Conference,* Neural Networks, Pages:1147 – 1156.

Kohonen, T., (2000). Self-Organizing Maps, Springer Series in Information Sciences, Springer, Berlin, Germany.

Mirkin, B.G., (1996). Mathematical classification and clustering

Ng, Y.S., and Srinivasan, R., (2004a). Monitoring of distillation column operation through self-organizing maps, *7th International Symposium on DYnamics and COntrol of Process Systems (DYCOPS)*, Massachusetts, USA.

Ng, Y.S., and Srinivasan, R., (2004b). Transitions in the Process Industries: Opportunities and Prospective Solutions, presented in IEEE International Symposium on Intelligent Control (ISIC), Taipei, Taiwan, Sep 2-4.

Nimmo, I., (1995). Adequately address abnormal operations, *Chemical Engineering Progress*, September 1995.

Schalkoff, R, (1992). Pattern recognition, Statistical, Structural and Neural Approaches, John Wiley & Sons, Inc, USA.

Sneath, P. H. A. and Sokal, R. R., (1973). Numerical Taxonomy. Freeman, San Francisco, CA.

Srinivasan, R., Gopal, S., (2002). Extracting information from high-dimensional operations data using visualization techniques, *AIChe meeting*, Indianapolis, #271c.

Ultsch,A., Siemon, H.P., (1990). Kohonen's self organizing feature maps for exploratory data analysis, *Proceedings of International Neural Network Conference (INNC'90)*, Kluwer academic Publishers, Dordrecht, pp. 305-308.

TABLES AND FIGURES

Table 1: Standard operating procedures (SOP) for startup

Distillation column startup SOP	
1.	Set all controllers to manual
2.	Fill reboiler with liquid bottom product
3.	Open reflux valve and operate the column on full reflux
4.	Establish cooling water flow to condenser
5.	Start the reboiler heating coil power
6.	Wait for all of the temperatures to stabilize
7.	Start feed pump
8.	Activate reflux control and set reflux ratio
9.	Open bottom valve to collect product
10.	Wait for all the temperatures to stabilize

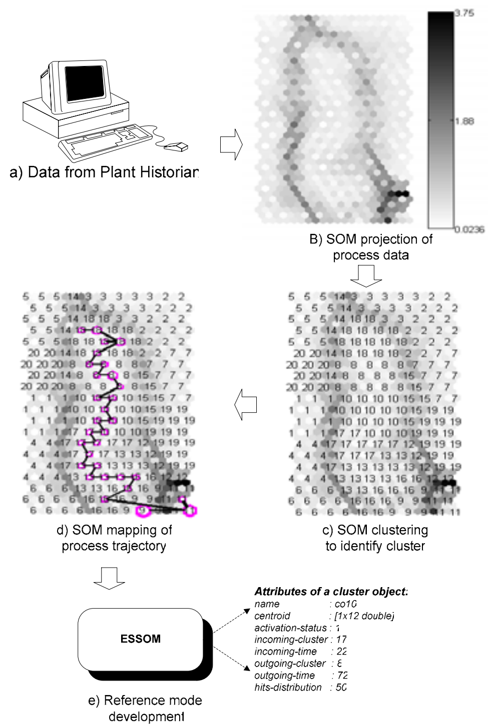

Figure 1: Abstraction of process data

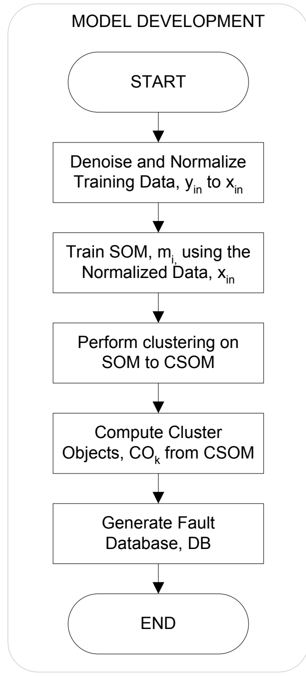

Figure 2: Offline training algorithm

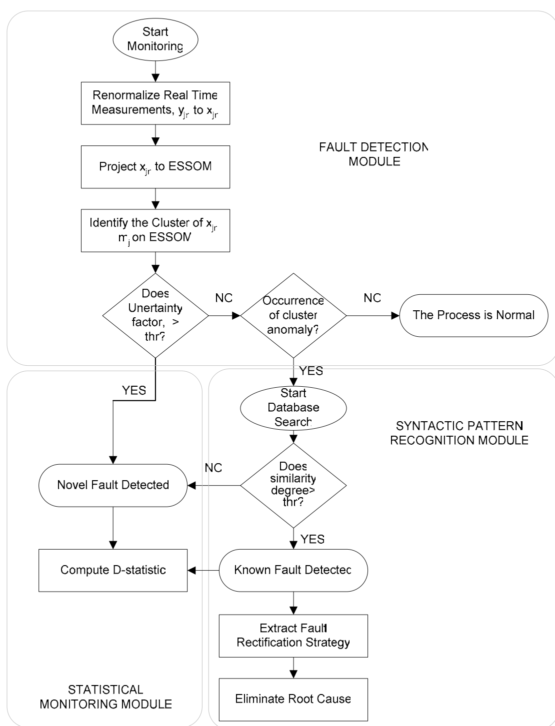

Figure 3: On-line fault detection and diagnosis algorithm

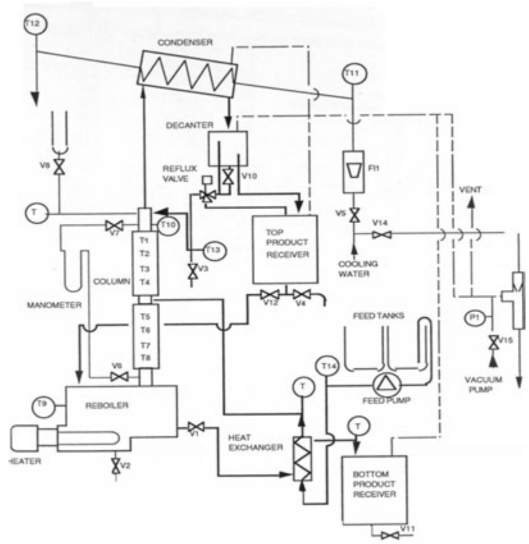

Figure 4: Schematic of the distillation unit set up

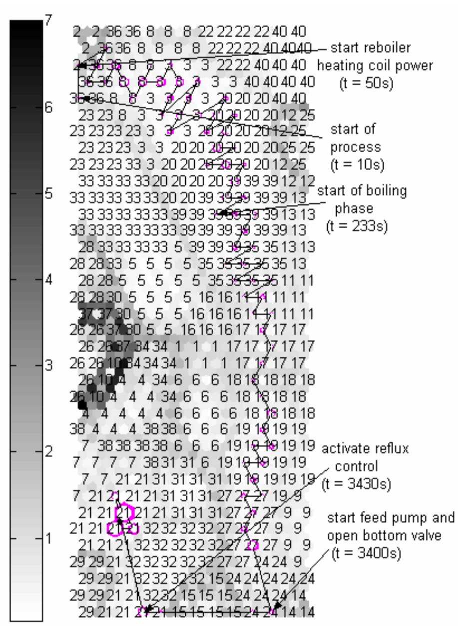

Figure 5: Trajectory of normal startup of distillation unit as projected on SOM U-matrix

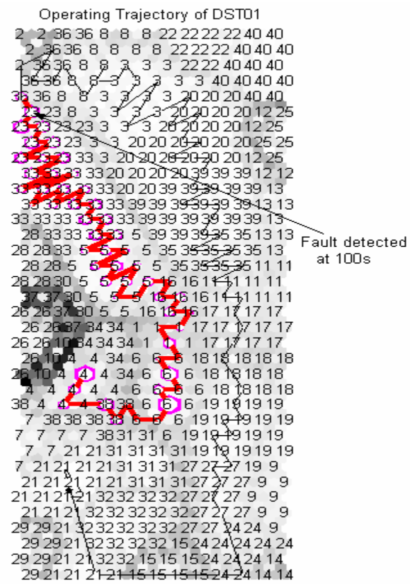

Figure 6: Operating Trajectory of DST01 (Reboiler power low)

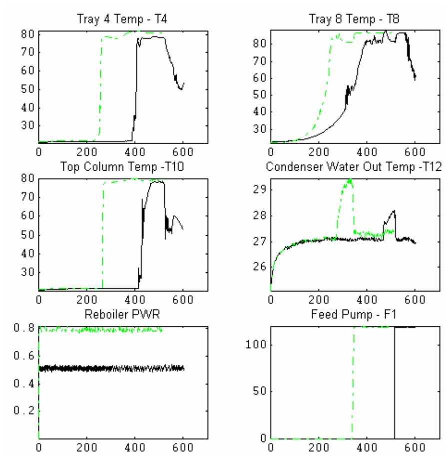

Figure 7: Process signals for DST01 (x10s)

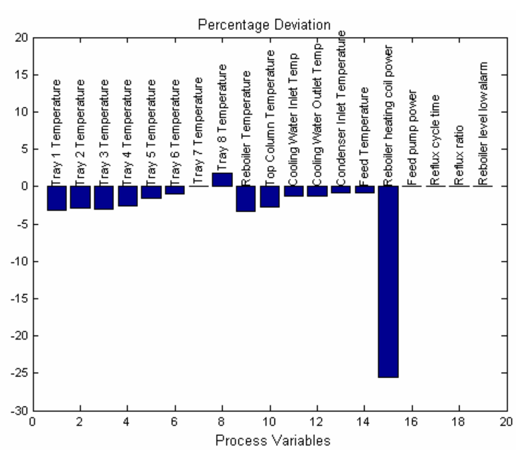

Figure 8: D-statistic contribution chart at time 100s for DST01