15th Triennial World Congress, Barcelona, Spain

# MAPREX: A SOM-BASED CONDITION MONITORING SYSTEM

**Abel A. Cuadrado** * **Ignacio Díaz** * **Alberto B. Diez** *
**Manuel Domínguez** ** **Juan A. González** *** **Faustino Obeso** ***

* *Universidad de Oviedo. Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas. Área de Ingeniería de Sistemas y Automática. Campus de Viesques s/n, 33204, Gijón, Asturias, Spain*
** *Universidad de León. Departamento de Ingeniería Eléctrica y Electrónica. Ingeniería de Sistemas y Automática. Escuela de Ingenierías Industrial e Informática. Edificio Tecnológico Campus de Vegazana. 24071, León, Spain*
*** *Centro de Desarrollo Tecnológico. Aceralia Corporación Siderúrgica. Apdo. 90. 33480 Avilés, Asturias, Spain*

Abstract: MAPREX is a software application for process condition monitoring based on the Self-Organizing Map (SOM) neural network. It provides advanced monitoring tools in a highly versatile way, allowing to look for the causes of a system malfunction or the factors which have an influence in the final product quality in complex systems as a previous stage for further more specific condition monitoring prototypes. Its main features include capability to acquire and store data from field sensors, real-time displaying of the signals and their spectra, real-time feature extraction and displaying, and state trajectory visualization based on SOM, also in real time. *Copyright © 2002 IFAC.*

Keywords: State Monitoring, Fault Identification, Neural Networks, Supervision

## 1. INTRODUCTION

In last years there has been a great effort in the industry oriented to the application of predictive techniques in maintenance of machinery and processes –good surveys can be found in (Gertler, 1988; Isermann, 1998; Patton and Lopez-Toribio, 1998; Betta and Pietrosanto, 2000)–. Supervision techniques have even gone beyond fault detection and isolation and aim also to monitor and predict the quality of the process performance. In many cases quality control techniques traditionally applied to the final product are being replaced by continuous quality monitoring and/or prediction based in on-line measurements of process variables and parameters by using data mining and visualization techniques –see e.g. (Whiteley and Davis, 1993; Alhoniemi *et al.*, 1999)–.

However, still in many cases, state of the art methods are insufficient to find out what is going on in a process. In complex processes –such as rolling mills, nuclear plants, etc.– previously unseen types of faults appear very frequently and require some *in situ* research by the technicians or engineers. This requires flexible and versatile tools, which allow to measure, *in situ* and simultaneously, variables in different test points and give as much information as possible to determine the problem. However, currently available portable tools –data loggers, oscilloscopes, spectrum analyzers– are too limited while, in turn, computer-based sophisticated systems are far from being portable.

This paper presents a portable condition monitoring system developed as a result of a research project for a Spanish steel company with powerful monitoring and intelligent data visualization methods based on the Self-Organizing Map. In the following sections the architecture of the developed system is described in detail as well as the visualization methods implemented.

## 2. SYSTEM DESCRIPTION

### 2.1 *Hardware description*

MAPREX was initially conceived as a portable tool for predictive maintenance. The system was implemented in a portable PC which contains a motherboard with disk units etc. and PCI slots allowing to use data-acquisition boards, as well as a built-in LCD display and a keyboard. All this material can be easily packed in a box of $40 \times 35 \times 25$ cm. The computer admits up to four data-acquisition boards which can acquire data simultaneously. There are several models available from Data Translation Inc., and the election of the exact model depends on the requirements in sampling frequency and number of signals. A photo of the whole system connected to acquire vibration data from a motor is shown in figure 1.

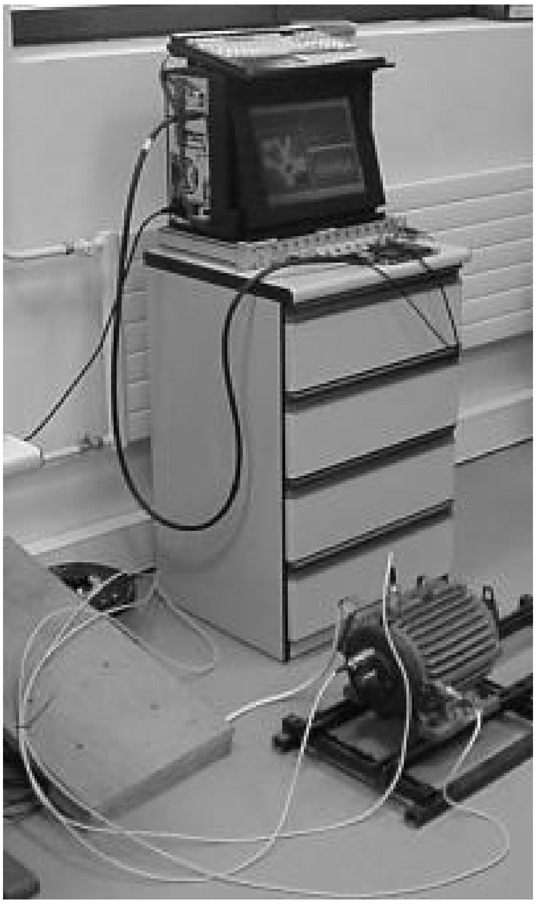


Fig. 1. The portable system configured to monitor vibrations of an asynchronous motor

### 2.2 *Software description*

The software application has three differentiated modules:

- Data acquisition module.
- Feature extraction module.
- SOM module.

*Data acquisition module.* The data acquisition module performs the tasks concerning acquisition, storage and displaying of data as temporal signals and its corresponding spectra obtained by means of Fast Fourier Transform (FFT). During the signal configuration the following properties can be changed: name, units and sensor sensibility. Each one of the acquisition boards has independent values in the sampling configuration: sampling frequency $f_s$, and sampling mode (temporal or spatial-angular). In this module, signals are presented in two modes: temporal mode, displayed in the *Time Window*, and frequency mode, displayed in the *Spectrum Window*.

*Feature extraction module.* The feature extraction module is useful to visualize the evolution in time of process features but it is also necessary as input stage to the SOM module. In this module some parameters of the STFT (*Short-Time Fourier Transform*) must be configured. These parameters are window size $L$, overlapping $M$, and window type (rectangular, Hanning, ...).

The signals for feature extraction must be chosen from those configured in the data acquisition module. Two types of signals are considered: *static* signals and *dynamic* signals.

*Static* signals are slow-varying signals –typically temperatures, pressures, etc.– whose wave parameters are not relevant. For this type of signals only their mean values are used as features.

*Dynamic* signals are fast-varying signals –typically vibrations, currents– whose waveforms need to be considered. For these signals, the energy/rms values contained at different frequency bands of the signal can be used as features. MAPREX allows two types of feature extraction for dynamic signals:

- *Fixed-frequency harmonics*. The energies contained in fixed-frequency bands are used as features by selecting the center frequencies $f_1, \cdots, f_n$ and bandwidths $B_{w1}, \cdots, B_{wn}$ for each feature.
- *Variable-frequency harmonics*. Many electrical and mechanical phenomena produce harmonics whose frequencies depend on other known variables. A typical case are rotating machines, which usually yield vibration harmonics at integer multiples $(1\times, 2\times, \cdots)$ of the rotation frequency $f_r$. If the rotation frequency of a machine is available as an analog measurement,

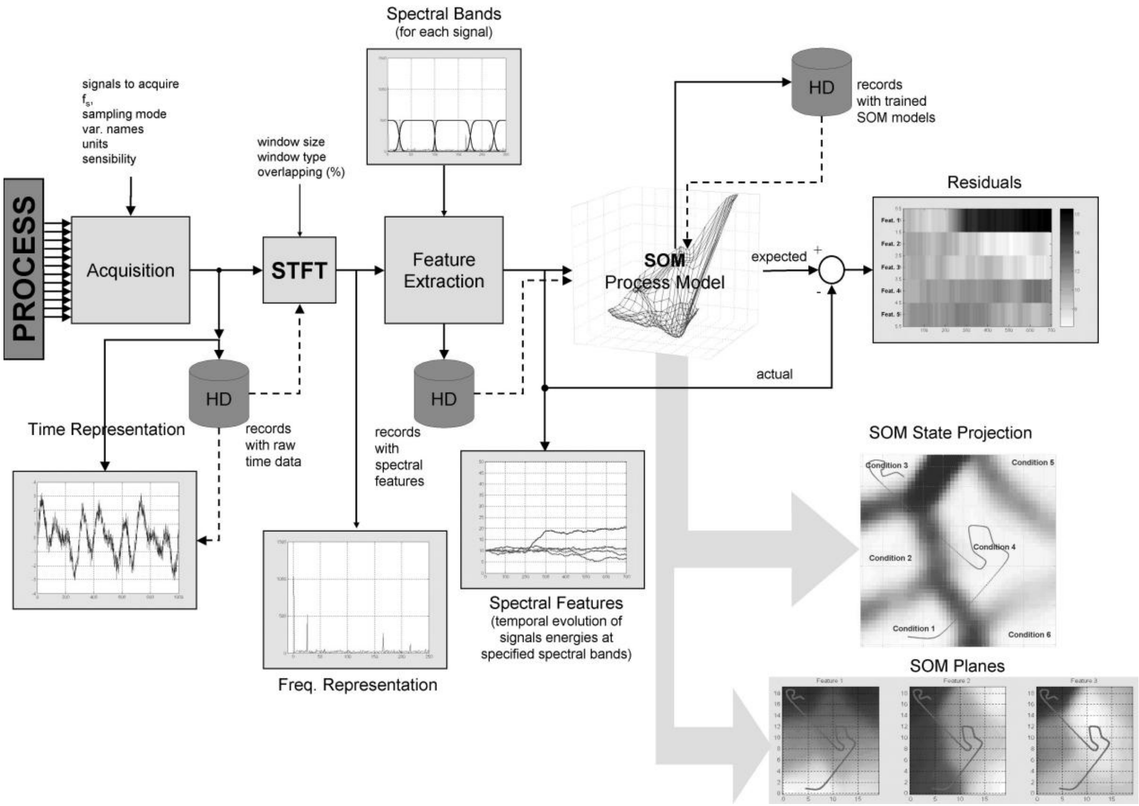



Fig. 2. General scheme of the MAPREX system.

$f_r$, then energies contained in bands around frequencies proportional to it, $K_1 f_r, \cdots K_m f_r$ can be tracked and used as features. This can be done in MAPREX by creating a list in a dialog box containing, for each feature: a) the measured analog variable $f_r$, to which the feature's frequency is related, b) the constant $K$, which relates the actual frequency of the feature to the measured variable, and c) the bandwidth $B_w$ around the variable center frequency, $K f_r$. This can be done for any number and type of “mother" frequencies –i.e. the $f_r$'s– and for any scalar multiples of them.

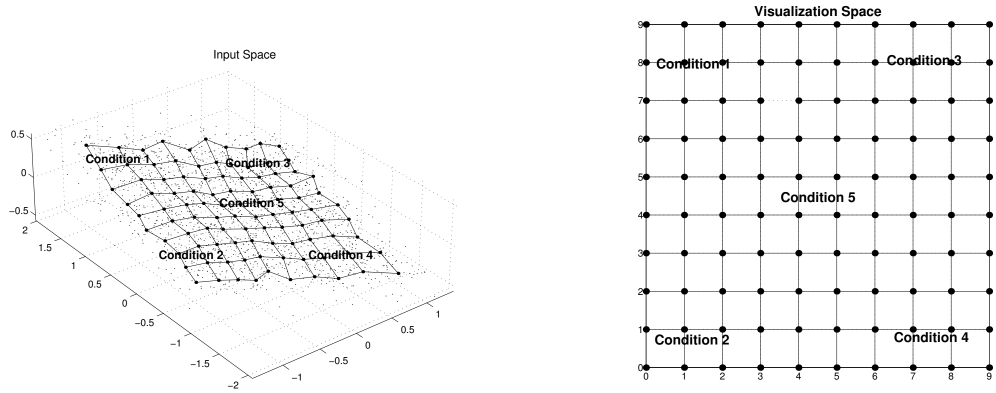
Finally, a normalization of the features can be enabled or not. During acquisition, the evolution along time of the configured features can be displayed in the *Feature Window*. The obtained feature vector can also be used as input to a Self-Organizing Map previously trained, as will be seen next.

*SOM module. Algorithm.* The feature vectors can reveal characteristic working conditions of the process in the form of data clusters in the feature space, since similar feature vectors correspond to similar working conditions. However, to distinguish many working conditions in complex processes many variables are needed. As a high-dimensional feature space cannot be graphically represented and the relationships between the process features usually are highly nonlinear, it is required a nonlinear projection method like the SOM to project feature space on a 2D space so that the data clusters corresponding to different working conditions do not overlap. In this manner the evolution of the process condition can be visualized as a trajectory of the instantaneous feature vector projected in this 2D space.

The Self-Organizing Map (Kohonen, 1995; Kohonen *et al.*, 1996), can be defined as a smooth nonlinear mapping from a high dimensional input space (here, the space of feature vectors) $\mathbb{R}^n$, onto a low dimensional (typically 2D) rectangular grid (also called *visualization space*). The map is defined by a set of points (*codebook vectors*) $\mathbf{m}_i$ in the input space and a corresponding set of nodes of a rectangular grid $\mathbf{g}_i$. A training algorithm (Kohonen, 1995) arranges the codebook vectors $\mathbf{m}_i$ so that they acquire the same geometry of the input data in a smooth and ordered fashion –see figure 3–.

For each feature vector $\mathbf{x}$, the SOM projection is defined so that the projection of $\mathbf{x}$ is the 2D node position $\mathbf{g}_c$ corresponding to the nearest codebook vector $\mathbf{m}_c$ to the actual feature vector $\mathbf{x}$ in the input space.

The SOM module includes training and execution submodules. The training submodule can train a SOM using a feature data set $\{\mathbf{x}_k\}$, which has been previously obtained and stored, using the *batch training algorithm* –see (Kohonen, 1995) for details on this




(a) Codebook vectors $\mathbf{m}_i$ in the input (feature) space.

(b) Regular SOM grid in the output (visualization) space formed by the nodes $\mathbf{g}_i$.

Fig. 3. SOM mapping. Note the matching of condition-related regions between both spaces.

algorithm–. For SOM training it is necessary to configure the SOM grid dimensions, $N_i, N_j$, the number of epochs $N$ and the file containing the feature data. These feature data have been previously produced by the feature extraction module through an off-line processing of stored process signal data. The SOM execution submodule performs the on-line projection of the instantaneous feature vector on the visualization space.

*SOM module. Planes.* The SOM projection algorithm explained above becomes extremely insightful by representing the so called *component planes* and *distance matrix*.

The *i*-th *component plane* (Kohonen *et al.*, 1996) is a 2D image built on the grid by considering each node (neuron) as a pixel with a color level proportional to the value of the *i*-th coordinate of the corresponding codebook vector. Indeed, if the *i*-th feature is, say, a temperature, the *i*-th component plane represents a temperature map of the process. The actual temperature of the process can be assessed just by looking the color below the actual projected point.

Other useful plane is the *distance matrix*. In the same idea as above, the mean distance of each neuron $\mathbf{m}_i$ to its topological neighbours in the grid is drawn. As neurons representing a cluster are very close and neurons outside a cluster are very sparse, this map reveals the cluster structure of the process data in 2D, and therefore, it shows up the different process conditions present in SOM training data.

Both representations can be shown simultaneously in different windows, each one along with its state trajectory projection.

*SOM module. Residuals.* Sometimes, the projection of the state trajectory with SOM can give misleading information about the process condition. Actually, the SOM manifold –the $\mathbf{m}_i$'s– adapts to feature data during its training phase and, during the exploitation stage, it only provides correct information if projected data are similar to those of training, i.e., if they correspond to conditions present in the SOM training data.

When a new condition –not present in the training set– appears in the process, the distance between the SOM manifold and the actual process feature vector $\mathbf{x}$ increases. The best matching –i.e. the closest– unit $\mathbf{m}_c$ is indeed the best approximation of the actual data $\mathbf{x}$ according to the learned SOM model. A quantity called *residual vector* (Díaz *et al.*, 2001):

$$\varepsilon(\mathbf{x}) = \mathbf{x} - \mathbf{m}_c \tag{1}$$

describes the deviation of the process with respect to the SOM model componentwise, showing up which variables deviate from their expected values and in which sense (by excess or by defect). As this vector depends also on time, a suitable 2D-image representation was chosen in which each pixel $(i, j)$ has a color value proportional to the $j$-th residual occurring at time $t_i$. Such image is shown in the *Residual Window*.

### 2.3 *Using the condition monitoring system*

In figure 4, a snapshot of the system is shown monitoring a 4 Kw asynchronous motor with three accelerometers, two Hall-effect current sensors and a PT-100 temperature sensor –the motor appears in the photo of figure 1–.

*Acquiring data.* To acquire data from the process, first, all the signals must be configured specifying names, units and sensor sensibilities. Then, the sampling frequency can be changed according to the bandwidth requirements of the signals. Data storage option

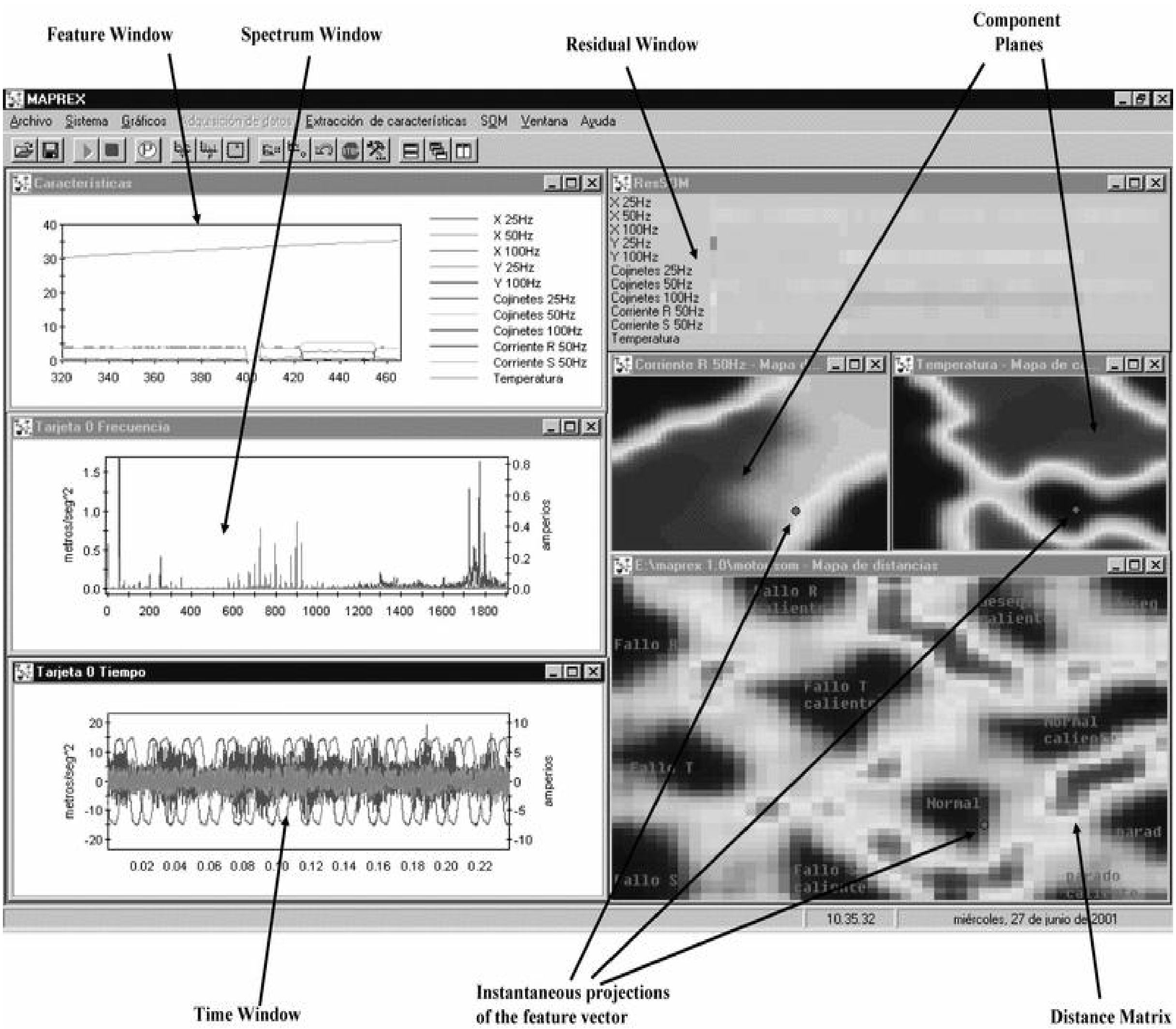



Fig. 4. A snapshot of the system.

and acquisition time must also be selected. With this minimal configuration the acquisition can be started. When the acquisition stops, the acquired data can be saved under any chosen file name and the acquisition procedure can be repeated. It is also possible to start acquisition automatically at programmed regular time intervals.

While MAPREX is acquiring (being the data storage option enabled or not) one can take a look at the Time and Spectrum Windows to observe the behavior of the process. The Feature Window is also available if the feature extraction is previously configured, with the spectral features selected as explained above. All these windows can help to select the feature set for a further analysis with SOM and residuals.

*Feature extraction and SOM training.* To obtain input data for SOM training, the feature extraction must be configured using the above procedure. Then, the system asks for process data files –previously acquired and stored–. Data sets should contain as many distinct working conditions of the process as possible. When the feature extraction is finished the feature extraction data can be saved in a file and it can be selected in the SOM training configuration. In addition, the SOM parameters (grid dimensions, training epochs) have to be specified. After SOM training, SOM data are saved.

*Exploiting the SOM.* Once a trained SOM is available, all kind of SOM windows can be opened: Component Planes, Distance Matrix, and Residual Window. If the acquisition has been started, the instantaneous projection of the feature vector is shown in each SOM window with a pointer.

The next analysis stage is to identify the process conditions corresponding to each region in the visualization space and to label them. A label placed in any SOM window (the Distance-Matrix Window, for instance) will appear also in the rest of the SOM windows (i.e., in all the Component Planes). Three possibilities are available to carry out this labelling:

- Labelled process data (separate data files each one belonging only to a known process condition) can be used to generate *activation maps*, which show the regions in SOM visualization

space corresponding to each file, i.e., to each known process condition. These *controlled samples* show where the labels must be placed and which are the labels to place.
- As mentioned in the section of the SOM module, the *distance matrix* is useful to distinguish regions corresponding to different conditions. It shows region boundaries and, therefore, it helps to place labels, but it gives no indication of which are the labels to place.
- The *component planes* can be used to estimate the value of process features, and therefore process variables, in each point of the SOM visualization space and hence being able to infer the process condition from them by applying expert knowledge about the process. They can be used in combination with distance matrix to know which and where the labels must be placed.

While the residuals are zero (green in Residual Window) the state indicated by the pointer is valid: the estimated values of the features in Component Planes and the cluster/condition pointed out by Distance Matrix are correct. However, it should be noted that, when at least one of the residuals deviates from zero, the information provided by the pointer only corresponds to the most similar condition present in the training data. Indeed, this means that the current process condition was not present in the SOM training data. If no conditions different from those of the training set are expected in the process, this can reveal the presence of incipient faulty states and the components of the residual vector can give an indication about the nature and origin of the new situation. At this point the user can choose to incorporate the new process data into the training set, and re-train the SOM, which will account for the new condition in subsequent executions.

## 3. CONCLUSION AND FUTURE WORK

MAPREX is a highly versatile portable system that represents an alternative to most of the commercial portable condition monitoring systems, which are typically meant for data logging, FFT monitoring and alarm threshold setting, and in most cases only allow a reduced number of signals to be analyzed simultaneously.

The system allows the technical staff to configure *in situ* the whole setup (measurement selection, feature extraction stage, etc.) according to the actual conditions of the plant and the production requirements. It can also be used just there where it is needed –in the same fashion as an oscilloscope– depending on the requirements of the plant maintenance, thus avoiding the need of installing many fixed CM systems. Other advantages can be outlined:

- High level of portability (can be carried with one hand).
- Can analyze a great number of signals simultaneously (up to 64 signals).
- On-line time or spectral representation of signals (live plots).
- Spectral feature extraction (track the spectral power content of any signal at configurable frequency bands).
- Learning ability. Spectral or time features of the process can be learned by a Self-Organizing Map (SOM).
- Graphical process visualization by SOM projection.
- Residual analysis. The difference between the learned model and the actual process features are highlighted graphically by means of the residuals.

Future steps will aim to improve its performance by allowing to use prior knowledge about the process. Currently, a fuzzy inference system is being implemented to aid in region identification and direct analysis of the features using fuzzy rules, which has yielded encouraging results in previous analysis and simulations.

## 4. REFERENCES


Alhoniemi, Esa, Jaakko Hollmén, Olli Simula and Juha Vesanto (1999). Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering* **6**(1), 3–14.

Betta, Giovanni and Antonio Pietrosanto (2000). Instrument fault detection and isolation: State of the art and new research trends. *IEEE Transactions on Instrumentation and Measurement* **49**(1), 100–107.

Díaz, Ignacio, Alberto B. Diez and Abel A. Cuadrado (2001). Complex process visualization through continuous self organizing maps using radial basis functions. In: *International Conference on Artificial Neural Networks (ICANN'01)*. Viena, Austria. pp. 443–449.

Gertler, J.J. (1988). Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine* **8**(6), 3–11.

Isermann, Rolf (1998). On fuzzy logic applications for automatic control, supervision, and fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics, Part A* **28**(2), 221–235.

Kohonen, Teuvo (1995). *Self-Organizing Maps*. Springer-Verlag.

Kohonen, Teuvo, Erkki Oja, Olli Simula, Ari Visa and Jari Kangas (1996). Engineering applications of the self-organizing map. *Proceedings of the IEEE* **84**(10), 1358–1384.

Patton, R.J. and C.J. Lopez-Toribio (1998). Artificial intelligence approaches to fault diagnosis. In: *IEE Colloquium on Update on Developments in Intelligent Control (Ref. No. 1998/513)*. p. 312.

Whiteley, James R. and James F. Davis (1993). Qualitative interpretation of sensor patterns. *IEEE Expert* **8**(2), 54–63.