International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-3, July 2011

Classification of Faults in DAMADICS Benchmark Process Control System Using Self Organizing Maps

Tarun Chopra, Jayashri Vajpai

***Abstract*—This paper presents a new approach for classification of faults in a process control system with complex overlapping fault classes. It is based on the application of Self Organising Maps that possess the capability of efficient unsupervised learning. Using the SOM training process, the proposed approach derives a set of neurons by considering process monitoring dataset comprising of multiple measured attributes. This set of neurons constitutes the multilayered SOM, in which each neuron corresponds to a class of faults. The neurons with similar attribute values are spatially arranged in adjoining localities, to set up an exploratory linkage between the SOM and the fault dataset. The performance of the proposed method is found to be satisfactory for fault diagnosis in the DAMADICS Benchmark Process Control System, even for the overlapping fault classes that pose considerable difficulty to other classification approaches applied by researchers.**

***Index Terms*—Artificial Neural Networks, DAMADICS Benchmark Process Control System, Fault Diagnosis, Self Organizing Maps.**

I. Introduction

Scientific research has recently focused on exploratory data analysis, in order to reveal the real knowledge embedded in large volumes of data. The automated analysis and visualization of massive multi-dimensional datasets has, hence, formed an important aspect of research. The principal objective is to find regularities and relationships in the data, thereby gaining access to hidden and potentially useful knowledge. Artificial Neural Networks have established their credentials as a promising tool for this purpose.

Kohonen's Self-Organizing Map (SOM) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called ‘map’. SOMs are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space so that similar data items will be mapped to nearby locations on the map [1].This special property allows creating spatially organized “internal representation” of various features of input signals and their abstractions. This makes SOM a powerful visualization tool for tasks which require processing of large amount of numerical data. Visual understanding of processes is facilitated by SOM; make it a useful tool for analysis, monitoring and modeling of industrial processes.

In this paper, SOM has been employed as a tool for the visualization of information related to fault diagnosis in Development of Applications and Methods for Actuator Diagnosis in Industrial Control Systems (DAMADICS) Benchmark Process Control System .This Benchmark is concerned with applications of process control and fault diagnosis methods on chosen actuators in the 5-stage evaporisation plant used in the Lublin sugar factory, Poland [2].

II. State of Art

A large number of faults commonly occur in the actuator valve block of a sugar plant during the dynamic production process. Further, there is a possibility that these may manifest with different levels of strength in two broad categories, i.e., abrupt {small, medium, big} and incipient. The early diagnosis of faults, which includes detection, identification and isolation, helps to minimize the associated damage in the industrial set up. DAMADICS provides a benchmark for the development and testing of process control and fault diagnosis methodologies. The available literature clearly depicts that DAMADICS Actuator, shown in Fig1 has been used by a considerable number of researchers [3] as a benchmark for evaluating their complex system modeling and fault diagnosis methodologies.

Structural analysis is among the most popular analytical fault diagnosis approaches attempted in process automation and has also been used on this benchmark [4], for early detection of faults and determination of their possibilities of detectability and isolability; Passive robustness problem in fault detection using interval observers has been studied by Puig et al [5]. Supavatanakul et al [6] have investigated the problem of fault diagnosis in discrete-event systems represented by timed automata. The application of a ‘‘signal-model’’-based fault detection method using squared coherency functions has been demonstrated by Previdi et al. [7].

Soft Computing based approaches that have been applied to this problem by various researchers include:

- Fuzzy classifier for fault detection and fault isolation based on particle swarm

Manuscript received June 26, 2011.

Tarun Chopra, Ph.D. Scholar, Department of Electrical Engineering, Faculty of Engineering, J.N.V. University, Jodhpur (India)-342 011 (email: tarun_ecb@rediffmail.com)

Dr. Jayashri Vajpai, Associate Professor, Department of Electrical Engineering, Faculty of Engineering, J.N.V. University, Jodhpur-342 011 India (email: jvajpai@gmail.com).

optimization [8];

- Neuro-fuzzy modeling involving a hybrid combination of neuro-fuzzy fault identification and unknown input observers in the Neuro-Fuzzy and De-coupling Fault Diagnosis Scheme (NFDFDS) [9]
- Group Method of Data Handling (GMDH) neural networks for robust fault detection [10]
- Fuzzy qualitative simulation based fault detection and fault isolation algorithm [11]
- A Hidden Markov Model Approach for Fault Detection and Diagnosis [12]

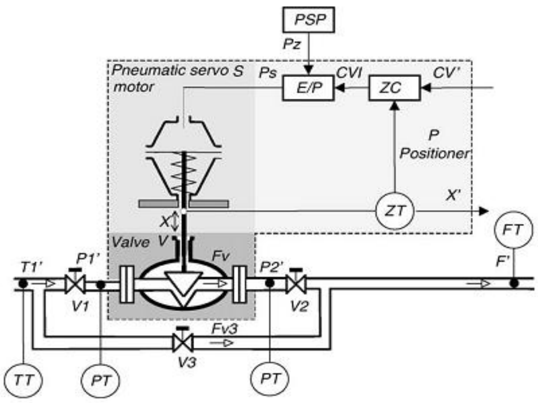

Fig. 1: Structure of Benchmark Actuator System [2]

However, the study of available literature has revealed that some of the DAMADICS faults are undetectable by the above mentioned approaches and cannot be isolated. In all, the benchmark system has provision for simulating 19 fault classes along with the normal operation. The datasets obtained from the simulation feature large overlapping areas in the classification. Consequently, the classification error is generally very large [4]. This indicates that further improvement is necessary and innovative techniques that enable better fault classification and their visualization should be developed.

Considering the above, an attempt has been made in this paper to apply SOM for better visualization of information related to fault diagnosis, their correlations and cluster structure of data in the DAMADICS Benchmark actuator. The Model developed in MATLAB- Simulink for this purpose is shown in Fig 2.

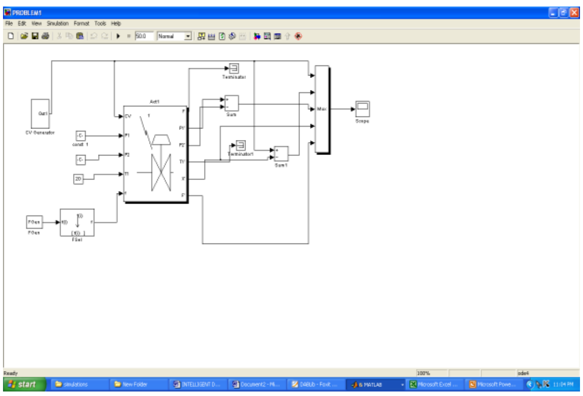

Fig 2: MATLAB - SIMULINK Model of the actuator

III. BASIC PRINCIPLES OF SELF ORGANIZING MAP

SOM tries to mimic the logical pattern of organizing information practiced by human brain, i.e., the connections of neurons within their logical group are much greater than their connections with the neurons outside the group. Thus, SOM appears to provide a more realistic model of certain aspects of human learning than many alternative neural models.

A. SOM Learning Algorithm

SOM follows the principle of unsupervised learning, i.e., it does not need a target output to be specified. Thus, wherever the node weights match the input vector, that area of the lattice is selectively optimized to more closely resemble the data for pertinent class of the input vector.

The SOM learning algorithm can be summarized in the following steps:

1. Initialization: - Choose random values for the initial weight vectors. The only restriction here is that w_j (0) should be different for j= 1, 2… l, where l is the number of neurons in the lattice. It may be desirable to keep the magnitude of the weights small.

2. Sampling: - Draw a sample x from the input space with a certain probability; the vector x represents the activation pattern that is applied to the lattice. The dimension of vector x is equal to m.

3. Similarity matching: - Find the best-matching (winning) neuron i(x) at time step n by using the minimum-distance Euclidean criterion:

$$i(x) = \arg(\min_j \|x(n) - w_j\|, j = 1, 2, \ldots, l$$

4. Updating: - Adjust the synaptic weight vectors of all neurons by using the update formula:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)(x(n) - w_j(n))$$

Where $\eta(n)$ is the learning-rate parameter, and $h_{j,i(x)}(n)$ is the neighborhood function centered around the winning neuron $i(x)$; both $\eta(n)$ and $h_{j,i(x)}(n)$ are varied dynamically during learning for best results.

5. Continuation: - Continue with step 2 until there are no noticeable changes in the observed feature map.

This procedure of self organizing and mapping as shown in Fig 3 allows for flexible descriptions, which results in the network topology alterations that lead to greater precision of the fault diagnosis.

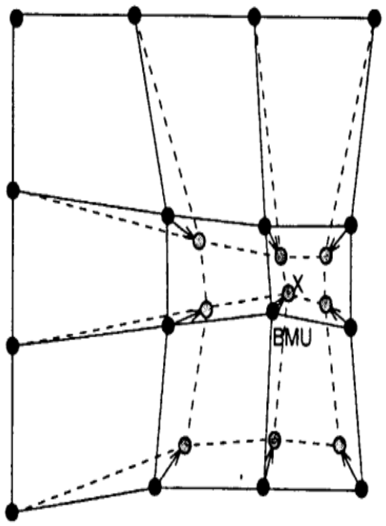

Fig. 3: Updating the BMU and its Neighbors [13]

(The solid and dashed lines represent to situation before and after updating, respectively)

B. Topology Preservation

Since not only the winning node is tuned towards the input pattern but also the neighboring nodes, it is probable that similar input patterns in future training cycles will find their best matching weight vector at nearby nodes on the map. In the run of the learning process, this leads to a spatial arrangement of the input patterns, thus inherently clustering the data. The more similar two patterns are, the closer their best matching units are likely to be on the final map [13].

C. Clustering of the SOM

When not enough labelled data is available, then, to facilitate analysis of the map and the data, similar units need to be grouped to reduce the number of clusters. This is due to the topological ordering of the unit maps. The clustered map can then be labeled. The primary benefit of this approach is to use more labeled data to assign each cluster a label and facilitate the analysis of revealed groups.

Davies–Bouldin index is a metric for evaluating clustering algorithms. It is a function of the ratio of the sum of within-cluster scatter to between-cluster separation.

$$DB = \frac{1}{n}\sum_{i=1}^{n}\max_{i\neq j}\left\{\frac{S_n(Q_i)+S_n(Q_j)}{S(Q_i,Q_j)}\right\},$$

where n - number of clusters, S_n - average distance of all objects from the cluster to their cluster centre, $S(Q_i Q_j)$ - distance between clusters centres. Hence the ratio is small if the clusters are compact and far from each other. Consequently, Davies-Bouldin index will have a small value for a good clustering. This index has been used in this paper.

D. SOM based Classifier

As the main purpose of fault diagnosis is to achieve an optimal mapping of the current state of the monitored systems onto a prespecified set of behaviors i.e. normal and faulty; hence it represents a suitable problem for classification methods. The SOM based classification is attractive, due to its unsupervised learning and topology preserving properties.

The SOM-based classifier separates object recognition into two parts:

1. Feature extraction with unsupervised learning in the first stage
2. Classification with supervised learning in the second stage.

An important guiding principle is that the features must be independent of class membership, since the latter is not yet known at the feature extraction stage by definition. This implies that if any learning methods are used for developing the feature extractors, they should be unsupervised in a sense, because the target class for each object is not known in advance. Fig 4 presents a typical SOM-based network.

The unsupervised learning stage is based on the rule that a neuron with the shortest Euclidean distance is fired while the others are inhibited from firing. The ranges of fired neurons vary as the learning process proceeds. The NN has the competitive relationship in a sense that once a neuron responds to a specific pattern, other neurons never reply to it. In other words, the input pattern that fires a certain neuron makes the neighboring neurons inactive.

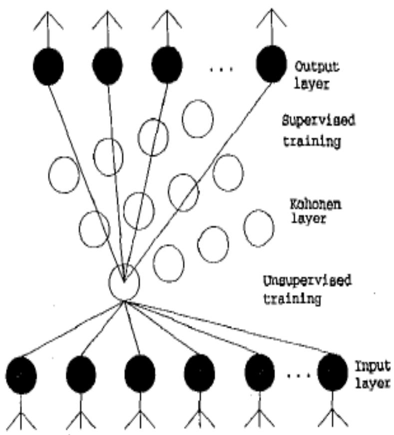

Fig 4: A SOM-based network structure [13]

IV. METHODOLOGY

The performance of SOM based fault classifier was tested for demonstrating its visualization abilities as follows: the sample fault data set pertaining to the fault classes F1, F2 and F3 has been chosen for this purpose. This group has been reported as belonging to the overlapping and indistinguishable faults category in [4]. It is analyzed using SOM Toolbox [14] in MATLAB environment.

A. Dataset Construction

The data set consisting of 50 representative samples from each of three types of faults (a total of 150 samples), is read from ASCII file. The measured variables are CV (process control external signal), P1 (pressure on valve inlet), P2 (pressure on valve outlet), X (valve plug displacement), F (main pipeline flow rate). The label associated with each sample is the fault type information viz 'F1' (valve clogging), 'F2' (valve or valve seat sedimentation) or 'F3' (valve or valve seat erosion).

B. Data Normalization

Since SOM algorithm is based on Euclidian distances, the scale of the variables is very important in determining the nature of map. If the range of values of some variable is much bigger than of the other variables, that variable will probably dominate the map organization completely. For this reason, the different components of the data set are usually normalized in the Pre-processing of Dataset, as shown in Fig 5.

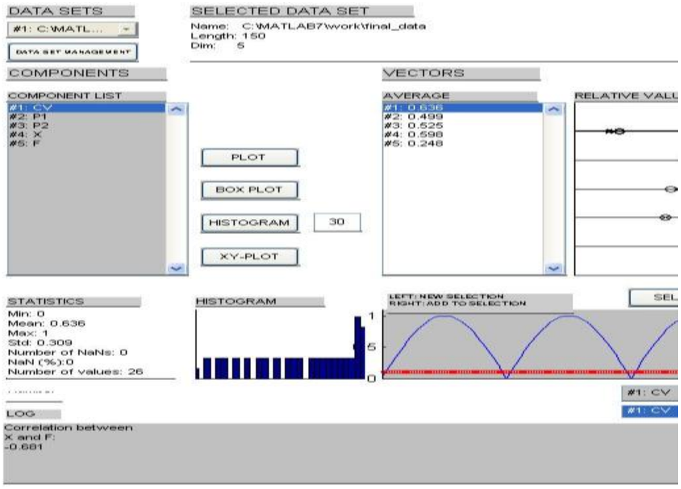

Fig 5: Preprocessing of Data Set

C. Map Training

The function SOM_MAKE is used to train the SOM. It first determines the map size, then initializes the map using linear initialization, and finally uses batch algorithm to train the map in following steps:

1. Determination of map size
2. Initialization
3. Training using batch algorithm
4. Rough training phase
5. Fine tuning phase

V. RESULT AND ANALYSIS

The qualitative and quantitative analysis of results obtained by using the proposed methodology is presented in following subsections:-

A. Map Training Results

The results for the chosen data set after Map training step are obtained as follows:-

- Map size = [11, 6] i.e., A two-dimensional SOM of 66 neurons (11 by 6), organized in a hexagonal neighborhood lattice.

After rough tuning phase, following results are obtained for the data set:-

- Quantization error: 0.827
- Topographic error: 0.060

Finally, after fine tuning phase, following results are obtained for the data set:-

- Quantization error: 0.791
- Topographic error: 0.053

B. Map Analysis by Visual Inspection

The first step in the analysis of the map is visual inspection. The U-matrix, component planes and labels obtained for the dataset are shown in Fig 6.

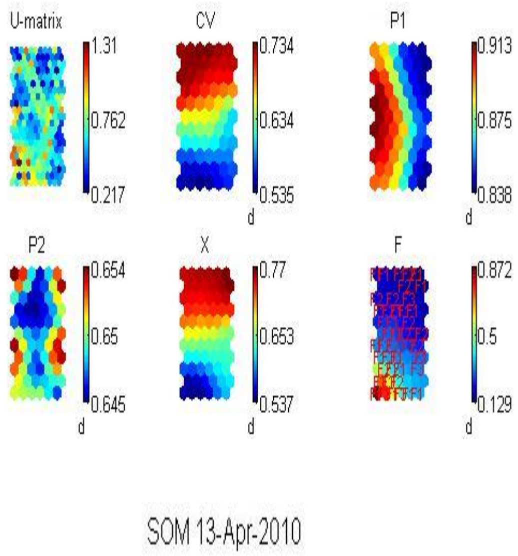

Fig 6: Visualization of U-matrix and Component Planes

The Unified distance matrix (U-matrix) is useful for detection of cluster borders and especially suitable for estimation of inter cluster distances. The U-matrix shows distances between neighbouring map units using color levels. Red color represents long distances and blue short ones. High values on the U-matrix mean large distance between neighboring map units, and thus indicate cluster borders. It is easy to see that the map unit in the top right corner is a very clear cluster. The U-matrix visualization has many more hexagons that the component planes. This is because distances between map units are shown and not only the distance values at the map units.

The SOM does not utilize class information during the training phase. Class labels can be displayed an empty grid as a post-process after the completion of training. Fig 6 clearly identifies the fault labels associated with each map unit (F1, F2, F3). From the labels it can be seen that unlabeled units indicate cluster borders and the map unit in the top right corner corresponds to the F1. The two other fault conditions form the other clusters. The U-matrix shows no clear separation between them, but from the labels it seems that they correspond to two sub clusters.

The component planes ('CV', 'P1', 'P2', 'X' and 'F') show values possessed by the prototype vectors of the map units. The value is indicated with color, and the color bar on the right shows what the colors mean. From Fig 6, it is clear that components CV & X are highly correlated.

The histograms and scatter plots of the five variables used in data set are shown in Fig 7.This visualization depicts quite a lot of information regarding distributions of single and pairs of variables both in the data and in the map. Original data points are shown in the upper triangle, map prototype values on the lower triangle, and histograms are shown on the diagonal. Black color has been used in the histogram for the data set and red for the map prototype values.

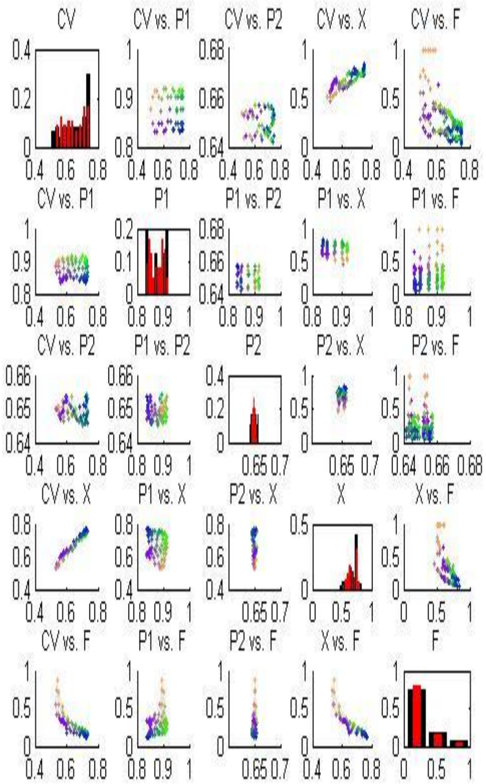

Fig. 7: Histograms and Scatter Plots

C. *Visualization of Projections*

Next, the projection of the data set has been investigated. A principal component projection is made for the data, and applied to the map. Distance matrix information is extracted from the U-matrix, and it is modified by knowledge of zero-hits (interpolative) units.

Finally, in Fig 8 three visualizations are shown: the color code (with clustering information and the number of hits in each unit), the projection and the labels. Neighboring map units are joined with lines to show the SOM topology. Labels associated with map units are also shown. These figures show that the projection confirms the existence of different clusters.

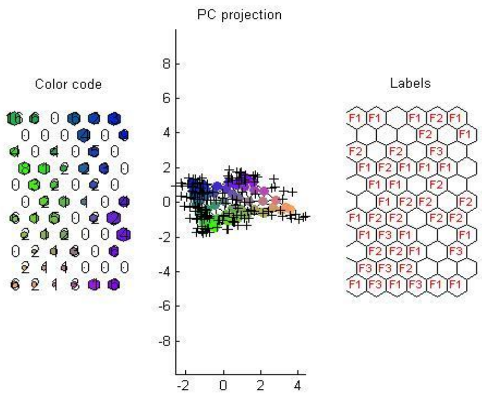

Fig 8: Visualization of Projections

D. *Clustering of the Map*

Visual inspection has already depicted that there are clusters in the data and the properties of these clusters are different from each other. For further investigation, the map needs to be partitioned i.e. to facilitate analysis of the map and the data; similar units need to be grouped to reduce the number of clusters. This is due to the topological ordering of the unit maps. Here, the KMEANS_CLUSTERS function has been used to find an initial partitioning. Fig 9 shows the Davies-Bouldin clustering index, which is minimized with best clustering.

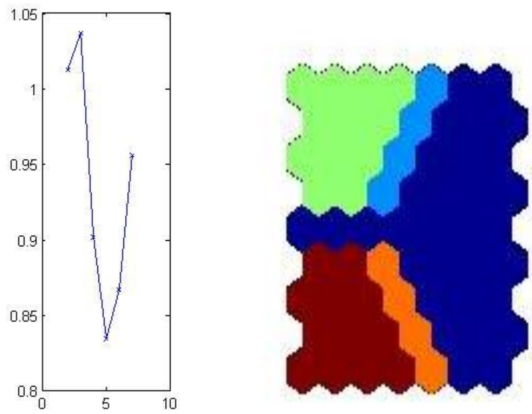

Fig 9: Clustering of SOM

The Davies-Bouldin index (0.95) obtained in this case, seems to indicate that there are chiefly three clusters on the map corresponding to faults F1,F2 and F3 with some percentage of overlapping.

E. *Classification*

Although the SOM can be used for classification, it is important to note that it does not utilize class information at all, thus, making its results inherently suboptimal. However, using function SOM_SUPERVISED, the network can take the class information into account.

After performing Classification task, following results are obtained for the data set:-

- Quantization error: 1.554
- Topographic error: 0.007

Consequent upon classification, U- Matrix obtained indicates clear cut separation between three categories of faults F1, F2 and F3 as shown in Fig 10.

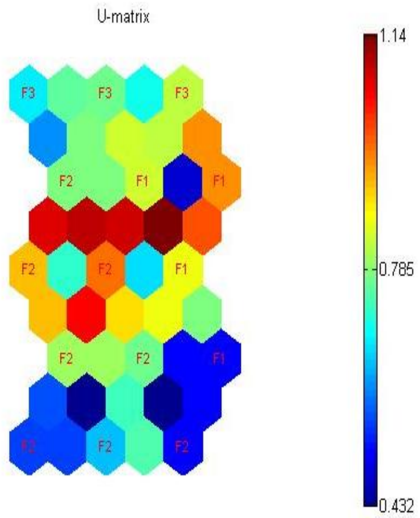

Fig 10: SOM after Supervised Learning

VI. DISCUSSIONS

In this paper, abilities of SOM have been demonstrated for fault detection and classifying the highly overlapping classes of actuator faults. Fairly comprehensible visualizations of SOM assist in monitoring the system health and may indicate which sensors respond to a particular fault and under what conditions. A fault may be indicated by a collective response of several sensors, which may not be obvious just looking at the data using other diagnostic techniques.

Visualization of changes in SOM over time may help greatly in the area of trend analysis. A suitable measure like quantization error can be defined, and if it increases beyond a certain bound for the new input vectors, an alarm can be raised to indicate that the system is approaching a faulty state.

Future research needs to focus on further improvement of fault diagnosis results on DAMADICS benchmark. One possible direction in which authors are presently working is to investigate the improvement in performance of the fault diagnosis task using perception based decision making.

REFRENCES

[1] T. Kohonen, "Self-Organizing Maps," Springer-Verlag Berlin, Heideberg, 1997.

[2] Michal Bartys et al, "Introduction to the DAMADICS actuator FDI benchmark study" Control Engineering Practice 14 , pp. 577–596, 2006

[3] Bocaniala et al, "On the applicability of state-of-the-art fault diagnosis methodologies to simple and complex systems", The annals of "DUNAREA DE JOS" University Of Galati Fascicle III, 2005 ISSN 1221-454X, Electrotehnics, Electronics, Automatic Control, Informatics

[4] Dustegor et al, "Structural analysis of fault isolability in the DAMADICS benchmark", Control Engineering Practice 14 , pp. 597–608, 2006.

[5] Puig et al, "Passive robust fault detection using interval observers: Application to the DAMADICS benchmark problem", Control Engineering Practice 14 , pp. 621–633, 2006.

[6] Supavatanakul et al, "Diagnosis of timed automata: Theory and application to the DAMADICS actuator benchmark problem", Control Engineering Practice 14 , pp. 609–619, 2006.

[7] Previdi et al , "Model-free actuator fault detection using a spectral estimation approach: the case of the DAMADICS benchmark problem", Control Engineering Practice 14 , pp. 635–644, 2006.

[8] Bocaniala et al , "Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem", Control Engineering Practice 14 , pp. 653–699, 2006.

[9] Uppal et al , "A neuro-fuzzy multiple-model observer approach to robust fault diagnosis based on the DAMADICS benchmark problem", Control Engineering Practice 14 , pp. 699–717, 2006.

[10] Witczak et al , "A GMDH neural network-based approach to robust fault diagnosis: Application to the DAMADICS benchmark problem", Control Engineering Practice 14 , pp. 671–683, 2006.

[11] Calado et al , "FDI approach to the DAMADICS benchmark problem based on qualitative reasoning coupled with fuzzy neural networks", Control Engineering Practice 14 , pp. 685–698, 2006.

[12] Almeida et al , "Fault Detection and Diagnosis in the DAMADICS Benchmark Actuator System –A Hidden Markov Model Approach" Proceedings of the 17th World Congress, The International Federation of Automatic Control,Seoul, Korea, July 6-11, 2008.

[13] Juha Vesanto,"Data Exploration Process Based on the Self−Organizing Map", Mathematics and Computing Series No. 115, 1999.

[14] Vesanto J., Himberg J., Alhoniemi E., Parhankangas J., "Self-organizing map in Matlab: the SOM Toolbox", In Proceedings of the Matlab DSP Conference 1999, Espoo, Finland, pp. 35-40, November 16-17, 1999.