PATTERN RECOGNITION TECHNIQUES USING FUZZILY LABELED DATA FOR PROCESS FAULT DETECTION

TEODOR MARCU*

The problem of robust diagnosis of process faults is addressed in the context of intelligent control. Fuzzy and non-parametric theoretic decision methods of pattern recognition are developed and applied to model-based fault detection based on parameter estimation. The present approach uses the concept of fuzzy sets to construct and to improve the performances of linear and nonlinear classifiers based on a distance. Simulation and experimental results regarding the diagnosis of various processes are included in two comparative studies.

1. Introduction

In order to achieve the total availability of processes, diagnosing systems based on either model or expert systems were developed (Frank, 1990; Isermann, 1991; Tzafestas, 1991). They enable a condition-based inspection and repair instead of applying "post-mortem" methods (Barschdorff, 1991). This is a response to the call for fault tolerance in automatic control systems (Frank and Köppen-Seliger, 1995). The active approach to fault tolerance provides fault accommodation, i.e. reconfiguration of the control system when a fault has occurred. For this purpose, a number of tasks has to be performed. The early diagnosis of the faults is one of the most important and difficult of them.

Recently, the approach of self diagnosing has become one of the main features of the so-called autonomous controllers (Aström, 1992; Isermann, 1992) and the general problem of fault detection and control was qualitatively stated by Narendra (1992). It illustrates best the key features of the modern technology of control engineering, well claimed as the acquisition of knowledge about systems and the processing of information about them (Trentelman and Willems, 1993).

In this context, the techniques of pattern recognition have a great impact on the stage of automatic acquisition of knowledge, learning and classification for the intelligent control of processes. As more intelligent control systems are designed, it becomes necessary to combine adaptation, learning and pattern recognition in novel ways to make decisions at various levels of abstraction (Narendra, 1992).

* Department of Automatic Control and Industrial Informatics, Technical University of Iasi, Str. Horia 7-9, RO-6600 Iasi 1, Romania, e-mail: tmarcu@ac.tuiasi.ro

The techniques used in fault diagnosis are mainly based on estimation methods, i.e. parameter and output variable estimation (Frank, 1990). These require mathematical models of processes. The model must represent the real plant satisfactorily and should not, however, be too complicated. This issue leads to the concept of robust fault diagnosis (Patton, 1994), due to the increasing demand for safe and reliable operation of uncertain and complex dynamic systems. In this respect, pattern recognition assures the so-called passive robustness in fault diagnosis, when the decision-making stage is made robust against uncertainty.

The widely used algorithms of pattern classification are statistical procedures (Willsky, 1986; Patton *et al.*, 1989), monitoring process coefficients (parameters) or states (outputs). However, they work under some restrictive hypotheses (Pao, 1989), e.g. some statistical distribution for patterns, one fault mode for each deviated parameter, etc. Moreover, the drawback of a high computational cost for complex problems of fault detection may affect the performances obtained, because the detection must occur with small delay and at fixed rates of false and missing alarms.

In order to overcome the above-mentioned limitations and to learn better discriminant functions for the purpose of classification, the paper presents a synthesis related to the integration of fuzzy and non-parametric theoretic decision techniques, i.e. patterns have here a free distribution in the feature space. This approach is based on the concept of treating the fault diagnosis as a problem of pattern recognition, on the one hand, and on the concept of fuzzy sets, i.e. a fuzzy weighted distance, fuzzy clustering and fuzzy linear regression, on the other hand. They are used to construct and to improve the performances of two types of trainable classifiers that are applied to fault detection. These are the linear classifier of minimum least-squares distance and the nonlinear classifier which is a multilayer feedforward perceptron with backpropagation learning rule. Thus, the present approach illustrates, in a new and unified framework, the concept of using fuzzily labeled data for the training portion of a classifier (Bezdek and Pal, 1992).

Two comparative studies of applying the presented techniques are included. They refer to the fault detection of a d.c. motor-pump-pipe process by means of numerical simulations, and to the diagnosis of the component faults in a laboratory process, namely the Three-Tank System. The investigations outline various aspects related to the design of the fault diagnosis schemes, in a close connection with the nature of the practical problems.

2. Basic Concepts and Tools

2.1. Problem of Pattern Recognition

The extracted features that form a pattern vector are considered to be either the process coefficients resulting from model parameters, or the parameters of process signals like frequencies, amplitudes, coefficients of linear predictors (Isermann, 1991), or residual signals given by an observer-based scheme (Frank and Köppen-Seliger, 1995). Thus, the process of feature selection means the use of an on-line method of parameter or output variable estimation.

In the following, one considers the known behaviour of a process characterized by many sets of total or partially different coefficients or parameters. They characterize the normal behaviour of the process and the faulty situations, if available. These data can be collected either at the process, if possible, or with the help of a simulation model that is as realistic as possible (Frank and Köppen-Seliger, 1995). The latter possibility is of special interest for collecting data of the different faulty situations since those data are not generally available at the real process.

The classifier is a decision-making device that is trained to classify an incoming pattern as belonging to one of the learned classes. Therefore, one considers the process parameters as an M-dimensional pattern, that is the symptom vector. We denote by $\{z_{kj}\}$, $j = \overline{1, N_k}$, $k = \overline{1, K}$ the sets of such patterns, where K is the dimension of the decision space and N_k is the number of patterns that belong to class k. They form the training set of the classifier, i.e. the knowledge basis:

$$\{z_l\}, \quad l = \overline{1, N}, \quad N = \sum_{k=1}^{K} N_k$$

For fault diagnosis, the task is to match each pattern of the symptom vector with one of the preassigned classes of faulty behaviour and the fault-free class (classes), respectively (Frank, 1994). This concept is illustrated in Fig. 1.
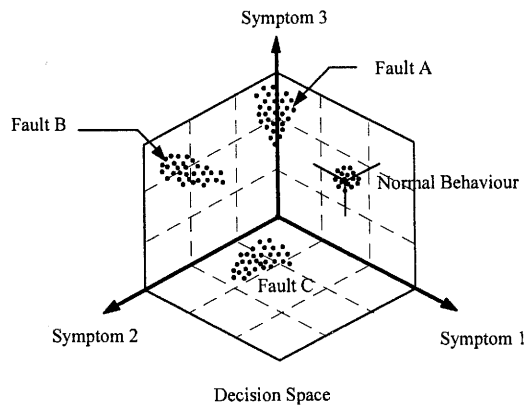

Fig. 1. Principle of the problem of fault classification.

When the normal behaviour of a process is characterized by many classes, e.g. the parameters may change slowly until the steady state occurs or may characterize some special classes of nonlinear models of a dynamic process, these classes must be recognized in a proper sequential order. Either the inobservance of this order or the failure in recognizing the model as one of the training sets of the classifier indicates a fault, too. In the latter case, the patterns that characterize the new faulty situation must be added to the training set and the synthesis of the decision mechanism must be reconsidered in an adaptive manner.

If properly chosen, the features produce closely arranged populations or clusters for each known class in the feature space (Barschdorff, 1991). Here, the variance and invariance properties of features play an important role in obtaining a robust classifier based on a distance. Moreover, in order to delete the disturbance of physical quantities and to take into account the parameter uncertainty (Frank, 1990), one must normalize the features (Dubes, 1992). However, this normalization must not alter the properties outlined above.

According to the principles stated above, the following *technique of normalization* is suggested (Marcu and Voicu, 1993):

N1. Determine the maximum value of each feature in the training set:

$$z_{\max}(m) := \max_{1 \leq l \leq N} \left\{ z_l(m) \right\}, \quad m = \overline{1, M}$$

N2. Scale each feature in the training set:

$$z_l^s(m) := z_l(m)/z_{\max}(m), \quad m = \overline{1, M}, \quad l = \overline{1, N}$$

N3. Determine the range of variation for each scaled feature and the maximum of them, respectively:

$$r(m) := \max_{1 \leq l \leq N} \left\{ 1 - z_l^s(m) \right\}, \quad m = \overline{1, M},$$

$$r_{\max} := \max_{1 \leq m \leq M} \left\{ r(m) \right\}$$

N4. Map each feature in a preselected interval $[a, b]$, according to the feature with the widest range of variation:

$$z_l^n(m) := b - \left[1 - z_l^s(m) \right] \left[(b - a)/r_{\max} \right], \quad m = \overline{1, M}, \quad l = \overline{1, M} \quad (1)$$

In this way, on the one hand, the disturbance of physical quantities is deleted, and, on the other hand, the feature properties of invariance and variance are enhanced.

Following this procedure, before classifying an unknown pattern $z \in \mathbb{R}^M$, we normalize it, according to the following relationship:

$$z^n(m) := b - \left[1 - z(m)/z_{\max}(m) \right] \left[(b - a)/r_{\max} \right], \quad m = \overline{1, M} \quad (2)$$

where $b, z_{\max}$ and $r_{\max}$ are determined in the training stage, as presented previously.

2.2. Fuzzy Sets

The concept of fuzzy sets can be used at the classification level for representing the class membership of objects. Here, the uncertainty in clustering or classification of patterns may arise from the overlapping nature of various classes. For fault diagnosis, this is a realistic situation, especially when incipient faults have to be detected.

2.2.1. Fuzzy Weighted Distance

Usually, the distances used in pattern recognition are of Minkowsky type, of which the Euclidean and Hamming distance are two special cases, the Mahalanobis distance, etc. One characteristic of these distances is that every feature of samples has the same effect on each distance. This is not exactly true in the real world, however (Lu and Fan, 1988; Bao-Wen, 1990). In this respect, it is necessary to weight the distance according to the different role of every feature of every sample, i.e. features are naturally and/or psychologically in different positions (Bao-Wen, 1990). This makes use of a weighted distance which is nearer to the way of people's thinking in recognizing things.

Thus, the fuzzy weighted Euclidean distance between two vectors z_i, $z_j \in \mathbb{R}^M$ is

$$d_{ij}^w := \|z_i^0 - z_j^0\|_2^w = \left\{ \sum_{m=1}^{M} c^m \left[z_i^0(m) - z_j^0(m) \right]^2 \right\}^{1/2}$$

where $c \in (0,1)$ is a controllable constant, z_i^0 and z_j^0 are the vectors z_i and z_j, respectively, reordered in accordance with the degree of importance of each feature, and $\| \cdot \|$ denotes a norm. For the particular case $c = 1$, all features have the same degree of importance.

Bao-Wen (1990) presented two kinds of distribution of importance degree in the selected features that arise physically (naturally) and psychologically. According to the former distinction, there are two principles to compare the importance of features that contribute to the clustering:

(A) The greater the value of the variance of a feature within a cluster is, the smaller the contribution of this feature to the clustering in this class becomes.

(B) The greater the value of the variance of a feature for all clusters is, the greater the contribution of this feature to the global clustering becomes.

Thus, principle (A) gives a modality of feature ordering characteristic to each class, and principle (B) gives a modality of feature ordering that allows for the discrimination between classes. These make the classification process nearer to the true links that exist between the objects of the real world and increase the separability of the classes.

As a result, the fuzzy weighted Euclidean distance presented previously is equivalent to the Euclidean distance of the vectors that have as elements (Marcu and Voicu, 1993)

$$z^{ow}(m) := \sqrt{c^m} z\left(i_{\mathrm{ord}}(m) \right), \quad m = \overline{1, M} \qquad (3)$$

where i_{ord} is the vector of feature positions, according to principles (A) and (B). The value 1 is assigned to the most important feature and the value M is assigned to the least important feature.

2.2.2. Fuzzy Clustering

In order to obtain automatically the classes in the training set and/or their charac-
teristics, the fuzzy ISODATA algorithm studied in (Kandel, 1982; Pao, 1989; Bezdek
and Pal, 1992) can be used, that is an unsupervised technique of clustering. This
procedure minimizes the sum of fuzzy variances of all features for each pattern in
each cluster, that is the minimization of the following cost:

$$J(F,V) := \sum_{l=1}^{N} \sum_{k=1}^{K} (\mu_{lk}^{\alpha} d_{lk}), \quad \alpha > 1 \tag{4}$$

where:

- F is the fuzzy partition, a K-tuple of the membership function $\{\mu_{lk}\}_{l=\overline{1,N};\, k=\overline{1,K}}$, denoting the membership degree of object l to the k-th class:

$$0 \leq \mu_{lk} \leq 1, \quad \sum_{k=1}^{K} \mu_{lk} = 1, \; \forall l, \quad 0 < \sum_{l=1}^{N} \mu_{lk} < N, \; \forall k \tag{5}$$

- $V = \{v_k, k = \overline{1,K}\}$, v_k—the prototype of the k-th class, that is the fuzzy mean;

- d_{lk} is the distance from the point z_l to the k-th class (with prototype v_k);

- α is a smoothing parameter that controls the "fuzziness" of the clusters. For the particular case $alpha = 1$, the clusters are separated by the so-called hard partitions. As α increases, the partitions become more fuzzy.

The minimization of the cost J is done with respect to (fuzzy) F, subject to $\alpha > 1$. Using the appropriate technique of normalization given by eqn. (1) and the concept of weighted distance presented in the previous sections, we can further improve this method, that is a challenge stated by Bao-Wen (1990). A revised form of this optimized algorithm (Marcu and Voicu, 1992; 1993), that is the *weighted fuzzy ISODATA*, contains the steps described in the sequel. *Th* denotes the threshold for the stopping criterion (step S5) and K_d is the desired number of classes (if not known, $K_d = N$).

For $K = 2$ to K_d do:

S1. *Initialization.* Choose a fuzzy partition $F = \{\mu_{lk}\}$ of non-empty membership degrees. The following initializations are now suggested:

  (a) the classes are not known:

    – assign uniform random values in (0,1) to $\{\mu_{lk}^i\}$;

– normalize according to eqns. (5):

$$s_l := \sum_{k=1}^{K} \mu_{lk}^i, \quad l = \overline{1,N}; \quad \mu_{lk} := \mu_{lk}^i / s_l, \quad l = \overline{1,N}, \quad k = \overline{1,K}$$

(b) the classes are known (K_d^i initial classes ω_k^i, $k = \overline{1, K_d^i}$):

$$\mu_{lk} := \begin{cases} 1 - (K_d^i - 1)\varepsilon, & z_l \in \omega_k^i \\ \varepsilon, & z_l \notin \omega_k^i \end{cases} ; \quad k = \overline{1, K_d^i}, \ l = \overline{1,N}, \ \varepsilon = 10^{-p}, \ p > 0$$

S2. Compute the feature values of cluster centres, i.e. the fuzzy means:

$$v_k(m) := \left[\sum_{l=1}^{K} \mu_{lk}^\alpha z_l(m) \right] / \left[\sum_{l=1}^{N} \mu_{lk}^\alpha \right], \quad m = \overline{1,M}, \quad k = \overline{1,K}$$

S3. Determine the feature positions:

S3.1. Compute the fuzzy variances of clusters:

$$s_k^f(m) := \left\{ \sum_{l=1}^{N} \mu_{lk}^\alpha \Big[z_l(m) - v_k(m) \Big]^2 \right\} / \left[\sum_{l=1}^{N} \mu_{lk}^\alpha \right], \quad m = \overline{1,M}, \quad k = \overline{1,K}$$

S3.2. Compute the global means and variances:

$$v(m) := \frac{1}{K} \sum_{k=1}^{K} v_k(m)$$

$$s(m) := \frac{1}{K-1} \sum_{k=1}^{K} \Big[v_k(m) - v(m) \Big]^2, \quad m = \overline{1,M}$$

S3.3. Compute the global function of feature ordering:

$$o(m) := \frac{1}{M} s(m) \sum_{k=1}^{K} \Big[1 - s_k^f(m) \Big], \quad m = \overline{1,M}$$

S3.4. Determine the feature positions: sort the vector o in descending order. The vector i_{ord} results and it contains the indexes used in sorting, that are the feature positions.

S4. Construct a new partition $\hat{F} = \{\hat{\mu}_{lk}\}$:

$$d_{lk}^w := \left\{ \sum_{m=1}^{M} c^{i_{\text{ord}}(m)} \Big[z_l(m) - v_k(m) \Big]^2 \right\}^{1/2}$$

$$\hat{\mu}_{lk} := 1 / \left[\sum_{q=1}^{K} (d_{lk}^w / d_{lq}^w)^{1/(\alpha-1)} \right], \quad k = \overline{1,K}, \quad \overline{1,N}$$

S5. Compute some convenient measure, δ, of the defect between F and $\hat{F}$, e.g. the Euclidean distance:

$$\delta := \left[\sum_{l=1}^{N} \sum_{k=1}^{K} (\mu_{lk} - \hat{\mu}_{lk})^2 \right]^{1/2}$$

If δ is less than a specified threshold Th, then go to step S6; else set $F := \hat{F}$ and go to step S2.

S6. Evaluate the criterion:

$$J_k^*(V^*) := \sum_{l=1}^{N} \left\{ \sum_{k=1}^{K} (d_{lk}^w)^{1/(1-\alpha)} \right\}^{1-\alpha}$$

where the asterisk denotes the optimal solution obtained in step S5.

The global optimal classification (K^* classes) is obtained according to the criterion

$$J_{K^*}^*(V^*) := \min_{2 \leq K \leq K_d} \left\{ J_K^*(V^*) \right\}$$

The classes are formed by selecting the patterns with the greatest membership grade corresponding to them.

2.3. Non-Parametric Theoretic Decision

Non-parametric classifiers map the patterns from the feature space into a decision space wherein the patterns belonging to a class are mapped to the cluster around a preselected, optimally chosen point, (Fu, 1980; Bow, 1992).

A common choice of the vectors of the decision space, $t_k \in \mathbb{R}^S$, $k = \overline{1,K}$, where S stands for the dimension of the decision space, is to consider $S = K$ and t_k as the vertices of K unit vectors:

$$t_k = [0 \ldots 0 \underset{k}{1} 0 \ldots 0]^\tau, \quad \|t_k\|_2 = 1, \quad k = \overline{1,K} \tag{6}$$

where τ denotes the transpose. These vectors have the property that the Euclidean distance between them is equal to $\sqrt{2}$, which ensures their linear separability (Sklansky and Wassel, 1981).

For a high value of K, another *choice of the vectors of the decision space* is considered (Marcu and Voicu, 1991), i.e. a subset of K vectors with S binary-valued components among the vertices of an S-cube:

$$t_k = \left[t_k(1) \ldots t_k(S-1) \, t_k(S) \right]^\tau, \quad t_k(s) \in \{0,1\}, \quad s = \overline{1,S}, \quad k = \overline{1,K} \tag{7}$$

$$2^{S-2} < K \leq 2^{S-1} \tag{8}$$

$$\overline{t_k(1)t_k(2) \ldots t_k(S-1)} = \text{the binary code of the decimal number } k-1 \tag{9}$$

$$t_k(S) = \begin{cases} 1 & \text{if } \displaystyle\sum_{s=1}^{S-1} t_k(s) = \text{an even number} \\[2mm] 0 & \text{if } \displaystyle\sum_{s=1}^{S-1} t_k(s) = \text{an odd number} \end{cases} \tag{10}$$

The K selected vectors have the main property that the Euclidean distance between them is greater than or equal to $\sqrt{2}$. Thus, the dimension of the decision space is evidently reduced from K to $S = 1 + [\log_2 K]$, where $[\cdot]$ denotes the function of the greatest integer.

2.3.1. Linear Classifier

The Least-Squares Minimum-Distance Classifier (LS MDC) is characterized by a transformation matrix $\boldsymbol{A}$ that maps the vectors of each class $\{z_{kj}\}_{j=\overline{1,N_k}}$ into a specified point t_k, $k = \overline{1,K}$, such that the overall mean-square error in the mapping is minimized (Fu, 1980).

For the k-th class, the least-squares condition is as follows: Minimize

$$E_k := \frac{1}{N_k} \sum_{j=1}^{N_k} E_{kj}^2 \tag{11}$$

$$E_{kj} := \|\boldsymbol{A} z_{kj} - t_k\|_2, \quad j = \overline{1,N_k}, \ k = \overline{1,K} \tag{12}$$

After taking the derivative of E_k with respect to the matrix $\boldsymbol{A}$, equating it to zero, generalizing it for all K classes and considering the augmented feature space, i.e. using the vectors, $\hat{z}_{kj} := [z_{kj}^{\tau} - 1]^{\tau}$ we obtain the following normal equations:

$$\boldsymbol{A} S_{\hat{z}\hat{z}} = S_{t\hat{z}} \tag{13}$$

where

$$S_{\hat{z}\hat{z}} = \sum_{k=1}^{K} \left[\frac{1}{N_k} \sum_{j=1}^{N_k} (\hat{z}_{kj} \hat{z}_{kj}^{\tau}) \right], \quad S_{t\hat{z}} = \sum_{k=1}^{K} \left[\frac{1}{N_k} \sum_{j=1}^{N_k} (t_k \hat{z}_{kj}^{\tau}) \right]$$

Equation (13) leads to

$$\boldsymbol{A} = S_{t\hat{z}} S_{\hat{z}\hat{z}}^{+} \tag{14}$$

where $^+$ denotes the generalized inverse of the involved matrix, determined through the analysis of its singular values (Golub and van Loan, 1989).

The concept of the LSMD classifier is illustrated in Fig. 2, where the classes are not linearly separable in the feature space and the linear mapping assures the separability in the decision space.

2.3.2. Neural-Network Classifier

The "supervised learning" of an Artificial Neural Network (ANN) performs inductive learning of a functional mapping from a feature space to a decision space, given a set of examples of instances of that mapping. In general, such a net is made up of nodes arranged in successive layers, as shown in Fig. 3.
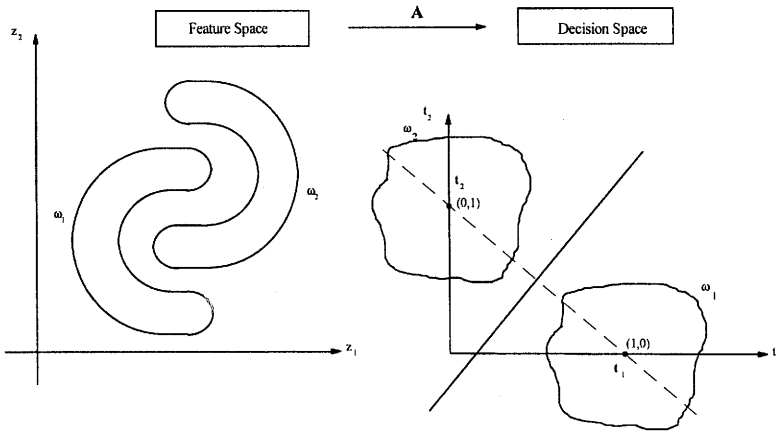

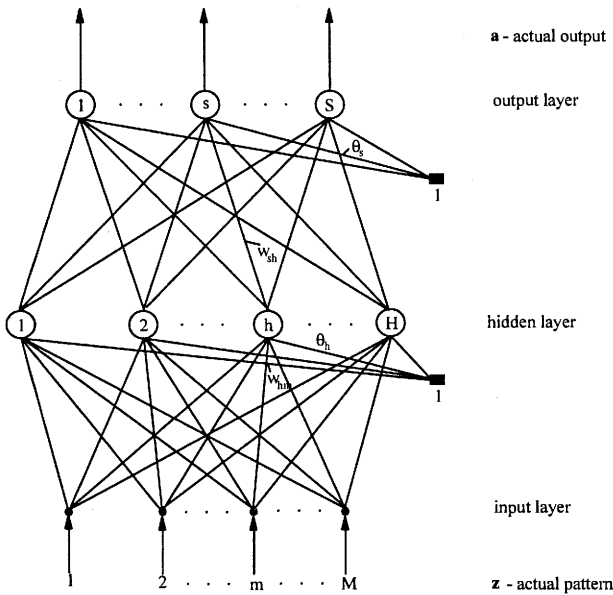Fig. 2. Least-squares minimum-distance classifier.

Fig. 3. Multilayer feedforward perceptron.

The successive layers of a MultiLayer feedforward Perceptron (MLP) with Back-Propagation (BP) learning rule carry out a sequence of mappings until one of them finds a representation in a suitable space, where the desired separation is possible. The well-known BP algorithm uses an iterative gradient technique to minimize the mean-square error between the desired outputs and the actual outputs of an MLP. This recursive algorithm starts at the outputs nodes and works back to the first hidden layer (Cichocki and Unbehauen, 1993). This network suffers from certain limitations (Pao, 1989). In particular, the learning rate is often too low to be acceptable.

Each input of the ANN, z_l, $l = \overline{1, N}$, must produce the desired output of the network $d_l \in T$, $l = \overline{1, N}$, $T = \{t_k; k = \overline{1, K}\}$. But z_l produces the actual output of the ANN $a_l = F(z_l)$. Here F is a vector function that is nonlinear and, in general, based on a nonlinear function of neurons, e.g. for the sigmoid (logistic) perceptron $f(x) = 1/(1 + e^{-x})$ (according to Fig. 3):

$$y_l(h) = f(net_h), \quad net_h = \sum_{m=1}^{M} w_{hm} z_l(m) + \theta_h, \quad h = \overline{1, H}$$

$$a_l(s) = f(net_s), \quad net_s = \sum_{h=1}^{H} w_{sh} y_l(h) + \theta_s, \quad s = \overline{1, S}$$

where y_l denotes the output vector of the hidden layer.

To learn this ANN means to solve the following problem:

given $\quad \{z_l\}_{l=\overline{1,N}}, \{d_l\}_{l=\overline{1,N}},$

find $\quad \{w_{hm}\}_{h=\overline{1,H}; m=\overline{1,M}}, \{\theta_h\}_{h=\overline{1,H}}, \{w_{sh}\}_{s=\overline{1,S}; h=\overline{1,H}}, \{\theta_s\}_{s=\overline{1,S}}$

that minimize

$$E := \sum_{l=1}^{N} \|d_l - a_l\|_2^2 \tag{15}$$

A solution is obtained by applying the approach of the steepest-descent gradient to minimize the mean-square error (15). This leads to the well-known generalized delta rule of backpropagation (Cichocki and Unbehauen, 1993):

$$\delta_{\text{output}}(s) = 2\Big[d_l(s) - a_l(s)\Big] f'(net_s), \quad f'(net_s) = a_l(s)\Big[1 - a_l(s)\Big] \tag{16}$$

$$\Delta w_{sh} = -\eta(\partial E/\partial w_{sh}) = \eta \delta_{\text{output}}(s) y_l(h), \quad \Delta \theta_s = -\eta(\partial E/\partial \theta_s) = \eta \delta_{\text{output}}(s) \tag{17}$$

$$w_{sh}^{\text{new}} := w_{sh}^{\text{old}} + \Delta w_{sh}, \quad \theta_s^{\text{new}} := \theta_s^{\text{old}} + \Delta \theta_s \tag{18}$$

$$\delta_{\text{hidden}}(h) = f'(net_h) \sum_{s=1}^{S} w_{sh}^{\text{old}} \delta_{\text{output}}(s), \quad f'(net_h) = y_l(h)\Big[1 - y_l(h)\Big] \tag{19}$$

$$\Delta w_{hm} = -\eta(\partial E/\partial w_{hm}) = \eta \delta_{\text{hidden}}(h) z_l(m), \quad \Delta \theta_h = -\eta(\partial E/\partial \theta_h) = \eta \delta_{\text{hidden}}(h) \tag{20}$$

$$w_{hm}^{\text{new}} := w_{hm}^{\text{old}} + \Delta w_{hm}, \quad \theta_h^{\text{new}} := \theta_h^{\text{old}} + \Delta \theta_h \tag{21}$$

where η denotes the learning rate.

The target vectors t_k, $k = \overline{1, K}$ of the decision space are those given by eqn. (6) or by eqns. (7)–(10), with their elements in the set $\{0.1, 0.9\}$ instead of $\{0, 1\}$. This choice takes into account the fact that the neurons in the output layer are of logistic type.

3. Fuzzy Linear Classifiers

3.1. Fuzzy Least-Squares MDC

To improve the performances of the LSMD classifier, we use a fuzzy technique of least-square mapping in the synthesis stage, i.e. the fuzzy linear regression (Jajuga, 1986). We reformulate the minimization problem stated by eqns. (11) and (12) by introducing the degrees of crisp membership of objects to the classes ω_k, $k = \overline{1, K}$:

$$u_{lk} := \begin{cases} 1, & z_l \in \omega_k \\ 0, & z_l \notin \omega_k \end{cases} \quad ; \quad l = \overline{1, N}, \quad k = \overline{1, K} \tag{22}$$

The mapping error corresponding to class k and given by eqn.(11) becomes

$$E_k := \frac{1}{N_k} \sum_{z_l \in \omega_k} u_{lk} E_{kl}^2$$

The optimization problem for this crisp case can be extended to the fuzzy case by drawing an analogy between the degrees of crisp membership $\{u_{lk}\}$ and those of fuzzy membership $\{\mu_{lk}\}$ that are introduced in Section 2.2, see eqns. (4) and (5). In this way, the transformation matrix A which maps $\{z_l\}_{l=\overline{1,N}}$ into the specified points t_k, $k = \overline{1, K}$ is obtained such that the overall fuzzy mean-square error in the mapping is minimized: Minimize

$$E^f := \sum_{k=1}^{K} \left[\frac{1}{N_k^f} \sum_{l=1}^{N} (\mu_{lk}^\alpha E_{kl}^2) \right], \quad N_k^f := \sum_{l=1}^{N} \mu_{lk}^\alpha$$

where N_k^f is the fuzzy number of patterns that belong to class k.

The matrices involved in the normal equations (13) and relationship (14) become

$$S_{\hat{z}\hat{z}}^f = \sum_{k=1}^{K} \left\{ \frac{1}{N_k^f} \sum_{l=1}^{N} \left[\mu_{lk}^\alpha (\hat{z}_l \hat{z}_l^\tau) \right] \right\}$$

$$S_{t\hat{z}}^f = \sum_{k=1}^{K} \left\{ \frac{1}{N_k^f} \sum_{l=1}^{N} \left[\mu_{lk}^\alpha (t_k \hat{z}_l^\tau) \right] \right\}$$

where $\hat{z}_l = [z_l^\tau - 1]^\tau$, $l = \overline{1, N}$.

Thus, we obtain the *Fuzzy Least-Squares Minimum-Distance Classifier* (FLSMDC, Marcu and Voicu, 1991). This method can be further improved (Marcu and Voicu, 1993), by using patterns that have normalized features, see (1), and after that, these are reordered and weighted, see (3). We obtain a new type of classifier, i.e. the *Fuzzy Least-Squares Minimum Weighted-Distance Classifier* (FLSMWDC).

3.2. Fuzzy Total Least-Squares MDC

The previous technique is reformulated in the Total Least-Squares (TLS) sense (Golub and van Loan, 1989) when having as vectors of the decision space those given by eqn. (6). For the k-th fuzzy class, the least-squares condition is to minimize

$$E_k^f := \sum_{l=1}^{N} \mu_{lk}^\alpha E_{kl}^2 = e_k^\tau F_k e_k = \| F_k^{1/2} (y_k - Z x_k) \|_2^2$$

where

$$y_k(l) = t_p(k), \quad l = \overline{\left(\sum_{j=1}^{p-1} N_j\right) + 1, \sum_{j=1}^{p} N_j}, \quad p = \overline{1,K}, \quad k = \overline{1,K}$$

$$Z = \begin{bmatrix} z_1 & \cdots & z_N \\ 1 & \cdots & 1 \end{bmatrix}^\tau, \quad F_k^{1/2} = \mathrm{diag}\left\{ \sqrt{\mu_{lk}^\alpha} \right\}_{l=\overline{1,N}}$$

The total fuzzy mean-square error in the mapping is of the form

$$E^f := \sum_{k=1}^{K} \frac{1}{N_k^f} E_k^f$$

The TLS solution, $(x_k)_{TLS}$, of the overdetermined system of linear equations

$$B[x_k^\tau - 1]^\tau = 0$$

is obtained by means of the singular value decomposition of the matrix (van Huffel and Vanderwalle, 1985):

$$B := \left[F_k^{1/2} Z \mid F_k^{1/2} y_k \right]$$

The matrix of the classifier is

$$A = [a_1 \ldots a_k \ldots a_K]^\tau, \quad a_k := \frac{1}{N_k^f} (x_k)_{TLS}$$

In this way, the perturbations in the "data" matrix Z are considered, and the separability of the classes is increased in the decision space. We obtain a new type of classifier (Marcu and Voicu, 1994), i.e. the *Fuzzy Total Least-Squares Minimum Weighted-Distance Classifier* (FTLSMWDC).

3.3. Fault Detection

Irrespective of the classifier used, to classify a given pattern z, it is first normalized, see (2), and then reordered and weighted, cf. eqn. (3). The vector z^{now} thus results. Thereafter, z^{now} is mapped into the decision space, and then classified as belonging to class k_0 if it is mapped closest to t_{k_0}, $k_0 \in \{1, \ldots, K\}$, according to the criterion

$$k_0 := \arg \min_{1 \leq k \leq K} \{d_k^2\}, \quad d_k = \|z^d - t_k\|_2, \quad z^d = A z^{\text{now}}$$

To verify if the recognized class is one of the training sets of the classifier, that is the concept of reject option (Dubuisson, 1992), the following decision criteria are applied sequentially:

(C1) $d_{k_0} \leq Th$

(C2) $d_{k_0} \leq Th_{k^*} (\leq Th)$

where k^* is the class of normal behaviour that must be recognized at a given instance of time, and the values of thresholds Th_{k^*} and Th are experimentally chosen, based on the training set of the classifier (Marcu and Voicu, 1993):

$$z_{kj}^d := A z_{kj}^{\text{now}}, \quad d_{kj,q} = \|z_{kj}^d - t_q\|_2, \quad k, q = \overline{1, K}, \quad j = \overline{1, N}$$

$$Th_k := \max_{1 \leq j \leq N_k} \left\{ \min_{1 \leq q \leq K} \{d_{kj,q}\} \text{ and } q = k \right\}$$

$$Th := \max_{1 \leq k \leq K} \{Th_k\}$$

The threshold values characterize the separability between the classes. A fault is detected if $k_0 = k^*$ and $d_{k_0} > Th_{k^*}$, or $k_0 \neq k^*$. The fault isolation is obtained if the recognized class corresponds to one of the learned classes that reflect a faulty behaviour, if available. If criterion (C2) is not satisfied for any known class or criterion (C1) is not satisfied, the pattern z belongs to an unknown (new) faulty class.

According to criteria (C1) and/or (C2), a pattern is situated in the decision space in the hypersphere k_0. In reality, this pattern is situated in the decision space in a hyperparallelepiped k_0:

(C3) $z_{\min, k_0}^d(m) \leq z^d(m) \leq z_{\max, k_0}^d(m), \quad m = \overline{1, M}, \quad k_0 \in \{1, \ldots, K\}$

where, for each class k, the following values are determined in the training stage:

$$z_{\min, k}^d(m) := \min_{1 \leq j \leq N_k} \left\{ z_{kj}^d(m) \right\}$$

$$z_{\max, k}^d(m) := \max_{1 \leq j \leq N_k} \left\{ z_{kj}^d(m) \right\}, \quad m = \overline{1, M}, \quad k = \overline{1, K}$$

Criterion (C3), if not satisfied, provides information about the features that are deviated in the case of a fault.

4. Simulation Results: Fault Detection of a D.C. Motor-Pump-Pipe Process

The diagnosis of a d.c. motor-pump-pipe system presented by Geiger (1985) and Isermann (1984; 1989) is studied through numerical simulation. We consider the model that consists of four nonlinear differential equations of first order that are linear in their parameters. The model is characterized by nine physical coefficients: L_a—the armature inductance, R_a—the armature resistance, ψ—the flux linkage, J—the momentum of inertia of the motor, K_f—the friction torque coefficient, the theoretic head coefficient, the head coefficient, the acceleration inverse coefficient and the coefficient of hydraulic resistance.

For the experiments, the armature voltage is changed stepwise. For the stage of feature selection, we consider the approach of Poisson Moment Functional (PMF) with minimal order for continuous-time models (Saha and Rao, 1983). This identification method is extended to the general form of linear/nonlinear state equations (Marcu and Voicu, 1991; 1994). The physical coefficients are obtained via nonlinear relations from the identified parameters of the mathematical model.

We consider the normal behaviour of the process characterized by many classes, knowing that a subset of the process coefficients varies after the start of the cold engine. R_a increases during, for example, 10 minutes (+10%), ψ decreases during 20 minutes (−4%), and K_f decreases during the first hour (−40%). In this case, the classes must be recognized in a proper sequential order. In the simulation task, some laws of variation are imposed on that set of coefficients (Isermann, 1984). For the rest of coefficients, a random uniform distribution around their nominal points of work is supposed. In the sequel, the identified coefficients are used in the classification task.

Firstly, the optimal number of classes for the normal behaviour of the system is identified. The knowledge base contains 100 samples for each set of physical coefficients resulted from identification repeated every 1 minute after the start of the cold engine. The clustering algorithm of fuzzy ISODATA is used (see Fig. 4).

The optimized algorithm of weighted fuzzy ISODATA indicates the optimal number of $K^* = 3$ classes of normal behaviour. The vector of features' position orders the elements of the pattern vector as follows: $K_f, R_a, \psi, \ldots$, the coefficient K_f being the most important feature. The third class (corresponding to the steady state) splits into two overlapped classes for $K = 4$. For $K > 4$, the supplementary classes have very few isolated elements or have no elements at all. The classical approach converges more slowly to the same solution, but without clear distinction between classes and with smaller values for the membership degrees. Thus, the optimized version of the clustering method with normalized inputs separates the classes better.

Secondly, the fault detection and isolation of the process is studied. It is based on the classes of normal behaviour resulting from as the outputs of previous experiments. Various types of faults of a d.c. motor-pump subsystem are considered here. They are caused by increased or decreased deviations of the process coefficients (up to 25% from the nominal values) from class 3 of the steady state.
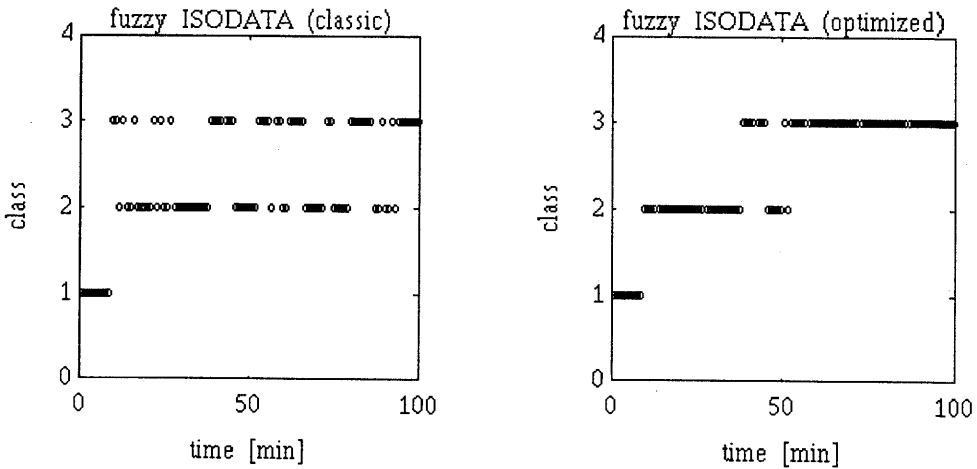
Fig. 4. Results of clustering of pattern data corresponding to the
normal behaviour of the d.c. motor-pump-pipe process.

The classification results of four types of classifiers are compared. They comprise
the multiple hypotheses GLLR algorithm that is widely used in fault diagnosis (Geiger,
1985; Isermann, 1989), the classical LSMDC, its fuzzy LS version, FLSMWDC, and
the fuzzy TLS version, FTLSMWDC. Table 1 presents the deviations from which the
faults are detected. For the last three types of classifiers, the classification results are
indicated using only the first two decision criteria, (C1) and (C2), and only the third
decision criterion, (C3), respectively.

Tab. 1. Results of fault recognition in a d.c. motor-pump subsystem: (∗) with
false alarms, (+) increasing deviations, (−) decreasing deviations.

Feature	Variation in class 3	GLLR	LSMD C1&C2	LSMD C3*	FLSMWD C1& C2	FLSMWD C3*	FTLSMWD C1&C2	FTLSMWD C3
$L_a(+)$	±1.0%	5.0%	—	4.0%	4.0%	3.0%	2.0%	1.5%
$R_a(+)$	±4.5%	5.5%	12.0%	5.0%	5.0%	5.0%	5.0%	5.0%
$\psi(-)$	±1.0%	12.5%	9.5%	8.0%	4.0%	2.0%	3.0%	1.5%
$J(-)$	±1.0%	5.0%	—	6.5%	3.5%	3.0%	2.0%	1.5%
$K_f(+)$	±3.0%	10.0%	—	—	8.0%	4.5%	3.5%	3.5%
$K_f(-)$	±3.0%	11.5%	20.0%	3.5%	4.0%	3.5%	3.5%	3.5%

A smaller delay of the alarm is remarked for the non-parametric classifiers. This is
due to the fact that the statistical method (GLLR) considers the previous estimates
of process coefficients that are undeviated together with the deviated ones, when
we compute the statistical parameters (means and variances) used in the decision

functions. Instead of this, the non-parametric classifiers consider only the current estimates.

We also remark upon the best results obtained with the FTLSMWDC. They are due to the use of normalized features, to the use of fuzzy weighted distance and, especially to the TLS approach. The LSMDC has some insensitivity to some faults and this is due to the fact that it is based on the classical distance, i.e. all features have the same degree of importance.

Moreover, the plots of the decision boundaries in the decision space together with the points corresponding to the artificial faults demonstrate the fact that the FTLSMWDC has a greater grade of class separability. This is shown in Fig. 5, where, the circles denote criterion (C2) and the rectangles denote criterion (C3).

The computational effort required in the stage of classifier design increases with the complexity of the classifier. The LSMDC requires fewer computations, while the FTLSMWDC is computationally more expensive. This aspect may be important when the decision mechanism must be quickly adapted to new (faulty) situations. On the other hand, the computational effort in the stage of active classification is the same for non-parametric classifiers. They have a reduced computational cost in comparison with the statistical procedure used. For the present application, the best solution to the problem of robust fault diagnosis is to use the FTLSMWDC.

5. Neuro-Fuzzy Learning

We reformulate the problem of ANN learning by using the degrees of crisp membership of objects to classes given by eqn. (22). Thus, the total error given by (15) takes the form

$$E := \sum_{k=1}^{K} \sum_{z_l \in \omega_k} u_{lk} \|t_k - a_l\|_2^2$$

By using the same analogy between the degrees of crisp membership and those of fuzzy membership, as stated in Section 3.1, and the previous notation and terminology, the fuzzy learning of the ANN means to solve the following problem:

given $\{z_l\}_{l=\overline{1,N}}, \{d_l\}_{l=\overline{1,N}}, d_l \in T = \{t_k; \; k = \overline{1,K}\}, \{\mu_{lk}\}_{l=\overline{1,N}; \; k=\overline{1,K}}$

find $\{w_{hm}\}_{h=\overline{1,H}; \; m=\overline{1,M}}, \{\theta_h\}_{h=\overline{1,H}}, \{w_{sh}\}_{s=\overline{1,S}; \; h=\overline{1,H}}, \{\theta_s\}_{s=\overline{1,S}}$

that minimize

$$E^f := \sum_{k=1}^{K} \sum_{l=1}^{N} \mu_{lk}^{\alpha} \|t_k - a_l\|_2^2 \tag{23}$$

If we denote by E_l^f the fuzzy error which is associated to the pattern z_l,

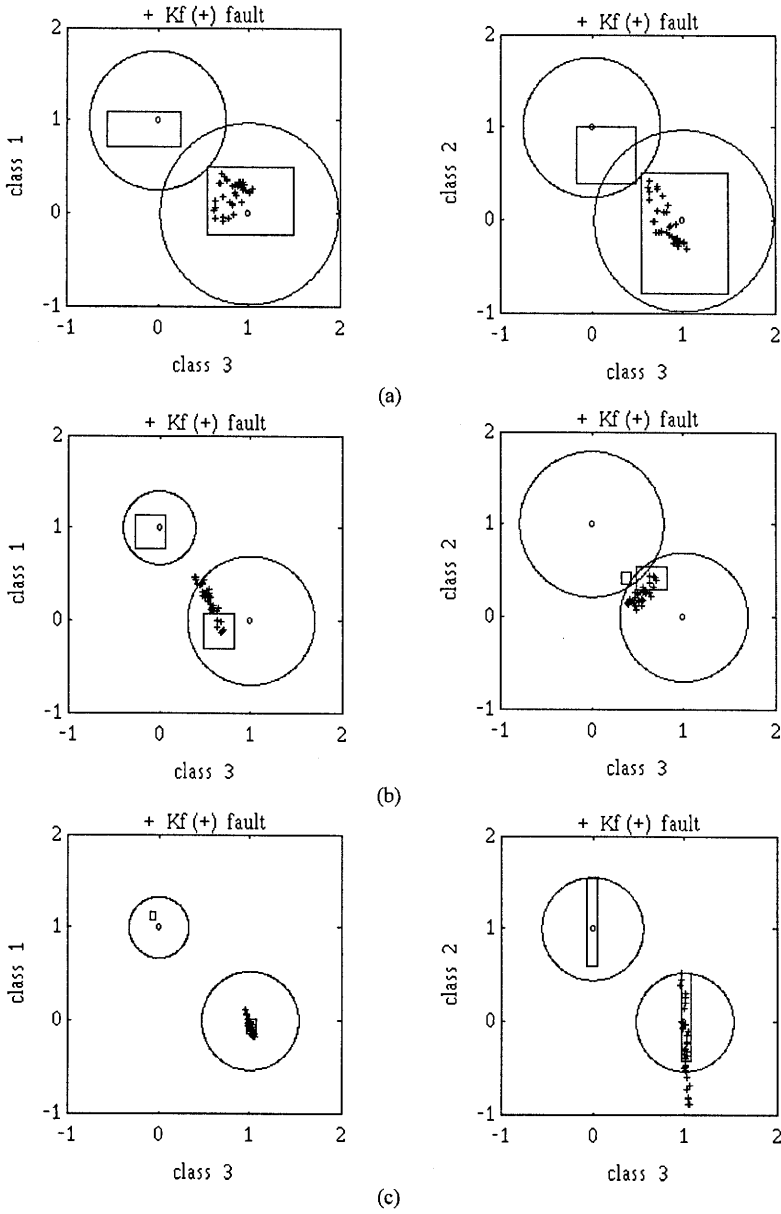$$E_l^f := \sum_{k=1}^{K} \mu_{lk}^{\alpha} \|t_k - a_l\|_2^2$$

Fig. 5. Decision boundaries corresponding to classes of normal be-
haviour of the d.c. motor-pump subsystem: (a) LSMDC,
(b) FLSMWDC, (c) FTLSMWDC.

the solution of the minimization problem for the fuzzy case is given by the *generalized delta fuzzy rule of backpropagation* (Marcu and Voicu, 1995). This rule affects directly only the weights and biases of the output layer. Thus, in eqns. (16)–(21) the generalized errors in the neurons, $\delta_{\text{output}}(s)$, are replaced by

$$\delta^f_{\text{output}}(s) := -\frac{\partial E^f_l}{\partial net_s} = 2 \left\{ \sum_{k=1}^{K} \mu^\alpha_{lk} \Big[t_k(s) - a_l(s) \Big] \right\} f'(net_s), \quad s = \overline{1, S}$$

The basic structure of the standard BP algorithm, eqns. (16)–(21), is preserved after modification of the L_2-norm criterion given with the use of the norm criterion given by (23), which is based on a type of non-Euclidean error (Cichocki and Unbehauen, 1993).

In the ideal case, when the output a_l equals its desired target, the optimal value of the error (23) does not become zero as for the crisp case. This is due to the fact that for a given class ω_k, we have

$$a_l = t_k \quad \text{and} \quad a_l \neq t_i, \quad i = \overline{1, K}, \quad i \neq k$$

An estimate of the optimal total fuzzy error is given by

$$E^{f^*} = \sum_{k=1}^{K} E^{f^*}_k, \quad E^{f^*}_k := \sum_{z_l \in \omega_k} \sum_{\substack{i=1 \\ i \neq k}}^{K} \mu^\alpha_{lk} \| t_k - t_i \|^2_2$$

The stage of fault detection/isolation can be implemented by using the criteria presented in Section 3.3. Moreover, if the target vectors of the MLP net are selected based on the analogy with the vertices of K-unit vectors, eqn. (6), and we have the elements in the set $\{0.1, 0.9\}$, the following mechanism of decision can be taken into consideration.

To classify a given pattern z, it is first normalized in a suitable manner. Thereafter, the resultant vector is mapped into the decision space, given the output of the ANN: $a \in \mathbb{R}^S$, $S = K$, and then classified as belonging to class $k_0 \in \{1, \ldots, K\}$, according to the criterion

$$k_0 := \arg \max_{1 \leq s \leq S} \Big\{ a(s) \Big\}$$

The concept of option rejection is implemented by applying the following criterion:

$$th_{k^*}(s) \leq a(s) \leq TH_{k^*}(s), \quad s = \overline{1, S} \tag{24}$$

where k^* is the class that must be recognized at a given instance of time, and the vectors th_{k^*} and TH_{k^*} respect the threshold values characterizing the separability of that class. These threshold values are chosen experimentally based on the training set of the classifier. For a given class ω_k, one can determine:

$$th_k(s) := \min_{\substack{z_l \in \omega_k \\ k^* = k}} \Big\{ a_l(s) \Big\}, \quad TH_k(s) := \max_{\substack{z_l \in \omega_k \\ k^* = k}} \Big\{ a_l(s) \Big\}, \quad k = \overline{1, K}, \ s = \overline{1, S}, \ l = \overline{1, N}$$

The following relations are usually satisfied: $th_k(s) \leq t_k(s) \leq \boldsymbol{TH}_k(s)$, where $t_k(s)$ are the elements of the target vectors. The threshold values characterize the quality of the training process, i.e. they must be as close as possible to the target elements.

Otherwise, a fault is detected. If the criterion (24) is not satisfied for any class, the pattern $\boldsymbol{z}$ belongs to an unknown class of the classifier. For this type of patterns, the synthesis of the classifier must be reconsidered together with the clustering procedure (Marcu and Voicu, 1995).

The concepts stated here can be applied to fault detection with linear classifiers that have as the vectors of the decision space those given by eqn. (6), too.

6. Experimental Results: Fault Diagnosis of a Three-Tank System

The experimental setup "Three-Tank System" (amira, 1993) consists of three cylindrical tanks (T_1, T_3, T_2) with identical cross sections filled with water. The tanks are interconnected by circular pipes. An appropriate stategy implemented on a microcomputer is used to control the water inlet by two pumps that are driven by an electronic power device. In this way, the volume flows of the lateral tanks T_1 and T_2 (the two process inputs) are controlled such that the level in the corresponding tanks (two out of three process outputs) can be preassigned independently of each other. The third output of the process, i.e. the level in the middle tank T_3, is always a resultant signal that is uncontrollable.

The diagnosis of component faults is studied using the Simulink/Matlab environment (MathWorks, 1992) and based on an additional block library that allows for the communication between the programme and the plant (Schönberger *et al.*, 1993). The real-time procedure is implemented in such a manner that the different stages of fault detection are conveniently overlapped with the control of the process.

The system can be modelled conveniently by the mass balances of the three tanks represented by three nonlinear differential equations of first order. For the purpose of fault diagnosis of process components, six physical coefficients that appear in the mathematical model are used. They are the cross-section areas of the tanks and the outflow coefficients, respectively. For the experiments, the reference values of the liquid levels are changed pulsewise with different magnitudes and durations for each controlled tank. The PMF approach with minimal order is used for feature selection. Further on, the identified coefficients are used in the classification stage.

The connecting pipes and tanks are additionally equipped with manually adjustable valves and outlets for the purpose of simulating clogs and leaks. Seven classes of process behaviour are taken into consideration. These are the normal behaviour (one class) and incipient faults, i.e. a leakage in each tank (three classes), clogging in the pipe between tanks (two classes), and clogging in the outlet of the process (one class). Thirty five experiments for each class were carried out in the period of a month, in order to take into consideration the influence of the plant environment. Table 2 presents an overview of the significant changes in the process coefficients, based on a statistical comparison of the extracted features. For each faulty class, a different pattern of changes is obtained.

Tab. 2. Effects of different faults on the estimated physical coefficients of a Three-Tank System: A_i–cross section area, az_i–outflow coefficient, T_i–tank i, $i = 1, 3, 2$, (+) increasing deviations, (−) decreasing deviations: 1–most deviated , ... , n–least deviated.

Fault type	A_1	az_1	A_3	az_3	A_2	az_2
L1: leakage in Tank 1	−(2)	+(1)	+(3)			
L3: leakage in Tank 3	−(3)		+(1)	−(2)	−(4)	
L2: leakage in Tank 2			+(3)	+(2)	−(4)	+(1)
C13: clogging pipe T1-T3		−(1)	+(2)	−(3)		
C32: clogging pipe T3-T2	+(3)		−(2)	−(1)	−(4)	
C20: clogging outlet T2	+(4)		+(3)	+(2)		−(1)

The complexity of the problem of pattern recognition is shown in Fig. 6, by the first and second principal components (Dubes, 1992), which together can explain 98.85% of the total variance of the pattern data. There are some clusters that are badly overlapping and some that are separated from other clusters.
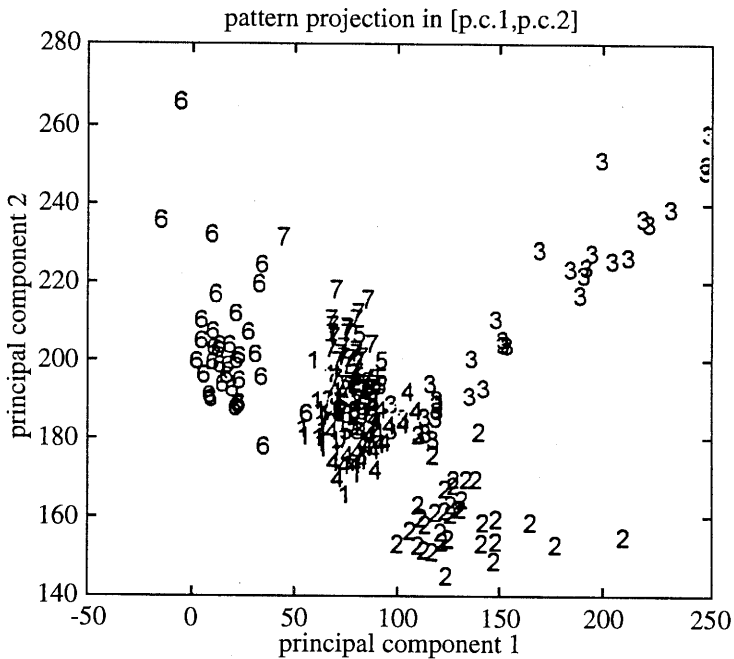

Fig. 6. Principal component analysis of pattern classes of the Three-Tank System: 1–normal behaviour, 2–fault L1, 3–fault L3, 4–fault L2, 5–fault C13, 6–fault C32, 7–fault C20.

The fuzzy ISODATA algorithm is first used to extract the most representative patterns from the available ones. It recognizes 90% of the patterns as belonging to the original classes. Further on, for each class, 25 out of 35 patterns, which have the greatest membership grades, are selected for the training set of different classifiers. The rest of patterns form the test set of classifiers.

The following classifiers are used and their results are compared: the statistical procedure GLLR, and the non-parametric classifiers. They are the classical LSMDC, the FLSMWDC, the FTLSMWDC, the MLP net with one hidden layer, the BP rule and the fuzzy BP rule (fBP), respectively. The fuzzy ISODATA algorithm is then employed to obtain the membership grades of patterns in the training set used further for the synthesis of fuzzified classifiers. In this case, the recognition rate of the clustering method is 100%. Table 3 presents the recognition rates of each type of classifier used.

Tab. 3. Recognition rates of fault detection in the Three-Tank System.

Recognition rate [%]	GLLR	LSMDC	FLSMWDC	FTLSMWDC	MLP/BP	MLP/fBP
training set (1)	—	95.42	96.00	97.14	100.00	99.42
test set (2)	—	61.42	67.42	72.00	88.57	84.25
global set (1)+(2)	66.66	85.71	87.75	89.94	96.73	95.02

We remark upon the best results obtained with the nonlinear classifiers, MLP/BP and MLP/fBP. However, the price to be paid is a greater computational cost involved in both stages of design and application of these classifiers. On the other hand, the net that uses the fuzzy rule has a greater rate of learning, as shown in Fig. 7. This is due to the fact that the fuzzy BP rule controls better the process of learning by means of membership degrees of objects to classes. In this way, a reduced time of learning is achieved. As mentioned in Section 4, this is of particular interest for the retraining of the decision mechanism.

Table 4 presents the recognized faults in the components of the Three-Tank System, corresponding to the training and test sets of the nonlinear classifier MLP/fBP.

In comparison with these results, the previous approach to the robustness problem, based on a bank of unknown input fault detection observers (Wünnenberg, 1990), detected only accentuated faults corresponding to mass flows of about 40 ml/s, due to the inherent limitations of the mathematical model used.

The diagnosis subsystem was tested for the situations corresponding to multiple faults, as well. Fifteen double faults were considered, i.e. all possible distinct combinations of two out of six faults presented previously. The decision mechanism indicates first one of two sources of the double fault. The correct decision obtained is accompanied by a reject alarm, the corresponding point in the decision space being situated in the neighborhood of one of the true faulty classes. The second source of fault was then recognized, after the elimination of the first fault.
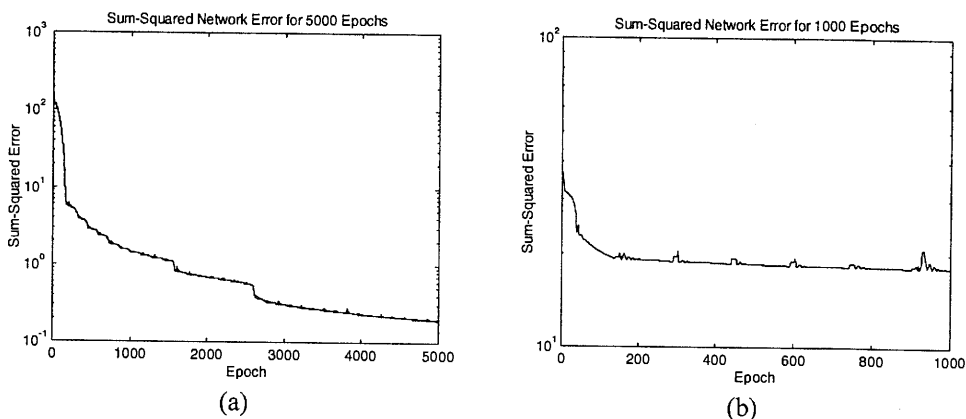
Fig. 7. Learning for the MLP net with one hidden layer: (a) BP learning rule, (b) fuzzy BP rule of learning.

Tab. 4. Recognized faults in the components of the Three-Tank System.

Fault type	Mass flow [ml/s]		
	minimum	mean	maximum
L1	9.15	14.24	23.82
L3	8.04	16.18	20.48
L2	9.06	18.44	28.72
C13	7.08	8.71	10.05
C32	9.95	18.99	26.45
C20	7.10	13.43	23.27

7. Conclusions

Today, Artificial Intelligence (AI) approaches to fault diagnosis in combination with analytical techniques form a basis for the design of control systems that are fault tolerant (Frank and Köppen-Seliger, 1995). In this respect, there is a growing interest in quantitative methods of AI applied to fault diagnosis, including the techniques based on pattern recognition and fuzzy sets.

The present approach diagnoses the fault by means of miscellaneous techniques based on the use of parameter estimation, and of fuzzy and non-parametric theoretic-decision methods of pattern recognition. The goal of assuring the robustness of the decision-making mechanism can be obtained if it is considered in all design stages of a system of pattern recognition dedicated to fault diagnosis, namely feature selection, feature analysis, cluster analysis and classifier design.

The present application examples illustrate the suggested theoretical developments and a systematic approach to the design stage. As a consequence, the best solution has to be picked up from the available ones, for each diagnosing system of a particular process. This must take into account the computational cost, too.

In this context, the presented techniques permit a better acquisition and understanding of the knowledge base on the one hand, and on the other, the detection with maximum sensitivity of any combination of deviated parameters, and the extension to the diagnosis of processes with many classes of known behaviour. They have also reduced the computational cost and, therefore, a smaller delay of the alarm results in comparison with statistical procedures.

The use of the concepts of fuzzy sets improves the performances obtained with the classical non-parametric methods, i.e. linear and nonlinear classifiers based on a distance. Thus, it makes the classification process nearer to the true way of people's thinking in recognizing things.

References

Amira (1993): *DTS200 Laboratory Experiment Three-Tank System.* — Duisburg, Germany: Amira GmbH.

Aström K.J. (1992): *Autonomous Controllers.* — Prep. IFAC/IFIP/IMACS Int. Symp. *Artificial Intelligenge in Real-Time Control,* Delft, The Netherlands, pp.1–6.

Bao-Wen L. (1990): *Weighted and graded fuzzy clustering.* — Fuzzy Sets and Systems, v.36, No.1, pp.37–43.

Barschdorff D. (1991): *Comparison of classical and neural decision functions.* — Prep. IFAC/IMACS Symp. *Fault Detection, Supervision and Safety for Technical Processes,* Baden-Baden, Germany, v.2, pp.429–430.

Bezdek J.C. and Pal K.S. (1992): *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data.* — New York: IEEE Press.

Bow S.T. (1992): *Pattern Recognition and Image Processing.* — New York: Marcel Decker.

Cichocki A. and Unbehauen R. (1993): *Neural Networks for Optimization and Signal Processing.* — Stuttgart: John Wiley & Sons & B.G. Teubner.

Dubes R.C. (1992): *Cluster analysis and related issues,* In: Handbook of Pattern Recognition and Computer Vision (C.H. Chen, L.F. Pau and P.S.P. Wang, Eds.). — Singapore: World Scientific Publ. Comp.

Dubuisson B. (1992): *An adaptive decision system using pattern recognition.* — Prep. IFAC/IFIP/IMACS Int. Symp. *Artificial Intelligenge in Real-Time Control,* Delft, The Netherlands, pp.667–670.

Frank P.M. (1990): *Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - A survey and some new results.* — Automatica, v.26, No.3, pp.459–474.

Frank P.M. (1994): *Application of fuzzy logic to process supervision and fault diagnosis.* — Prep. IFAC Symp. *Fault Detection, Supervision and Safety for Technical Processes,* Espoo, Finland, v.2, pp.531–538.

Frank P.M. and Köppen-Seliger B. (1995): *New developments using AI in fault diagnosis.* — Prep. IFAC/IMACS Int. Workshop *Artificial Intelligenge in Real-Time Control,* Bled, Slovenia, pp.1–12.

Fu K.S. (1980): *Digital Pattern Recognition.* — Berlin: Springer.

Geiger G. (1985): *Technische Fehlerdiagnose mittels Parameterschätzung und Fehlerklassifikation am Beispiel einer elektrisch angetriebenen Kreiselpumpe.* — Düsseldorf: VDI-Verlag GmbH, Reihe 8, Bd.91.

Golub G.H. and van Loan C.F. (1989): *Matrix Computations.* — London: John Hopkins University Press.

van Huffel S. and Vanderwalle J. (1985): *The use of total linear least-squares techniques for identification and parameter estimation.* — Prep. IFAC Symp. Identification and Parameter Estimation, York, UK, pp.1167–1172.

Isermann R. (1984): *Process fault detection based on modelling and estimation methods.* — Automatica, v.20, No.4, pp.387–404.

Isermann R. (1989): *Process fault diagnosis based on dynamic models and parameter estimation methods,* In: Fault Diagnosis in Dynamic Systems: Theory and Applications (R.J. Patton, P.M. Frank and R.N. Clark, Eds.). — London: Prentice-Hall Int. Ltd.

Isermann R. (1991): *Fault diagnosis of machines via parameter estimation and knowledge processing.* — Prep. IFAC/IMACS Symp. Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany, v.1, pp.121–133.

Isermann R. (1992): *Towards intelligent control of mechanical processes.* — Prep. IFAC/IFIP/IMACS Int. Symp. Artificial Intelligenge in Real-Time Control, Delft, The Netherlands, pp.7–23.

Jajuga J. (1986): *Linear fuzzy regression.* — Fuzzy Sets and Systems, v.20, No.3, pp.343–353.

Kandel A. (1982): *Fuzzy Techniques in Pattern Recognition.* — New York: John Wiley.

Lu Y. and Fan X.L. (1988): *Fuzzy weighted distance and its applications in pattern recognition.* — Proc. Int. Conf. Pattern Recognition, Rome, Italy, v.2, pp.1065–1067.

Marcu T. and Voicu M. (1991): *Miscellaneous pattern classification techniques for process fault detection and isolation.* — Prep. IFAC/IMACS Symp. Fault Detection, Supervision and Safety for Technical Processes, Baden-Baden, Germany, v.1, pp.175–180.

Marcu T. and Voicu M. (1992): *Fuzzy weighted clustering and classification for process fault detection.* — Prep. IFAC/IFIP/IMACS Int. Symp. Artificial Intelligenge in Real-Time Control, Delft, The Netherlands, pp.275–280.

Marcu T. and Voicu M. (1993): *Fuzzy and non-parametric techniques of pattern recognition for fault detection in dynamic systems.* — Proc. ECCA/IFAC 2nd European Control Conf., Groningen, The Netherlands, v.2, pp.457–502.

Marcu T. and Voicu M. (1994): *A total least-squares approach to pattern recognition for model-based fault detection.* — Prep. IFAC Symp. Fault Detection, Supervision and Safety for Technical Processes, Espoo, Finland, v.2, pp.702-707.

Marcu T. and Voicu M. (1995): *A neuro-fuzzy approach to pattern recognition for process change detection.* — Proc. EUCA/IFAC/IEEE 3rd European Control Conf., Rome, Italy, v.3, pp.1999–2004.

MathWorks (1992): *Simulink Users' Guide.* — Natick, MA: The MathWorks Inc.

Narendra K.S. (1992): *Intelligent control using neural networks.* — IEEE Control Systems, v.12, No.6, pp.11–18.

Pao Y.H. (1989): *Adaptive Pattern Recognition and Neural Networks.* — Reading, MA: Addison-Wesley.

Patton R.J., Frank P.M. and Clark R.N. (1989): *Fault Diagnosis in Dynamic Systems: Theory and Applications.* — London: Prentice-Hall Int. Ltd.

Patton R.J. (1994): *Robust model-based fault diagnosis: the state of the art.* — Prep. IFAC Symp. *Fault Detection, Supervision and Safety for Technical Processes,* Espoo, Finland, v.1, pp.1–24.

Saha D.C. and Rao G.P. (1983): *Identification of Continuous Dynamical Systems - The Poisson Moment Functional Approach.* — Berlin: Springer.

Schönberger F., Ifrim S. and Lazör C. (1993): *Simulink block library for real-time control.* — Prep. SEFI/TEMPUS Workshop Computer Science Topics for Control Engineering Education, Vienna, Austria.

Sklansky J. and Wassel G.N. (1981): *Pattern Classifiers and Trainable Machines.* — Berlin: Springer.

Trentelman H.L. and Willems J.C., Eds. (1993): *Essays on Control: Perspectives on the Theory and Its Applications.* — Boston: Birkhäuser.

Tzafestas S. (1991): *Second generation diagnostic expert systems.* — Prep. of IFAC/IMACS Symp. *Fault Detection, Supervision and Safety for Technical Processes,* Baden-Baden, Germany, v.2, pp.1–6.

Willsky A.S. (1986): *Detection of abrupt changes in dynamic systems,* In: Detection of Abrupt Changes in Signals and Dynamical Systems (M. Basseville and A. Benveniste, Eds.). — Berlin: Springer.

Wünnenberg J. (1990): *Observer Based Fault Detection in Dynamic Systems.* — Düsseldorf: VDI-Verlag GmbH, Reihe 8, Bd.222.