Asian Journal of Information Technology 5(7): 750-760, 2006

Neuro-Fuzzy Methods for Fault Diagnosis of Nonlinear Systems

Mehennaoui, L., [1]N. Debbache and M.L. Benlouci
Department of Electrotechnic, Skikda University, Algeria
[1]Department of Electronic, Annaba University, Algeria

Abstract: The study presents a Fault Detection and Isolation (FDI) scheme with a particular emphasis placed on sensor fault diagnosis in nonlinear dynamic systems. The non-analytical FDI scheme is based on a two-step procedure. Two methods are proposed for the first step, called residual generation, one use fuzzy sets and the second neuronal network. A fuzzy neural network performs the second step, called residual evaluation. Some simulation results are given for efficiency assessment of this fault diagnosis approach.

Key words: Fuzzy identification, neural identification, fault diagnosis, neuro-fuzzy scheme

INTRODUCTION

The problem of Fault Detection and Isolation (FDI) is a crucial issue for the safety, reliability and performance of industrial processes.

The FDI procedure consists basically of two main steps: generation of residuals which should be useful fault indicators and residual evaluation which involves decision making.

The model-based FDI approach also referred to as the analytical approach, which has received intensive attention, use mainly state and parameter estimation techniques[1-4]. The main drawback of the analytical approach is the requirement of an accurate model for reliable diagnostic decision (minimum rate of missed detections and false alarms).

A fundamental aspect in the design of model-based methods is thus concerned with the problem of robustness with respect to model uncertainties arising in the form of modelling errors and unknown external disturbances. As far as linear systems are concerned, the problem of robust residual generation may be considered to be mature[5,3,4] whereas the FDI problem for nonlinear dynamic systems has been investigated to a lesser extent[6,7].

Alternately, FDI can be performed using qualitative techniques such as expert systems, fuzzy logic, neural networks[8-15].

In a fault diagnosis procedure for linear systems used a combination of an analytical residual generator (a Kalman filter) and a fuzzy neural network for residual evaluation. This study extends this work to the nonlinear case. The main difference is the problem of the identification of non-linear model. On the other hand, we know now the capacity of fuzzy systems[16] and neural network[17] to identify nonlinear systems.

Once the model is obtained, a neural network performs the decision-making, which consists in detecting and isolating a fault when it occurs. This neural network coupled to a fuzzy inference block acts as an on-line fault classifier.

Residual generation: There are several different approaches to modelling of complex nonlinear systems. The main distinction can be made between global and local methods.

In this study, we present the two approaches: neural network in the global approach and fuzzy sets in the local one.

The residual generation procedure is depicted in Fig. 1.

Residual generation by fuzzy sets: The fuzzy sets methods use partitioning of the process domains into a number of fuzzy regions. For each region in the input space, a rule is defined that specifies the output of model. The rules can be seen as local of submodels of the systems. The rules used, here in this paper, are Takagi-Sugeno (TS) rules that give as result locally nonlinear submodels.

Takagi- Sugeno model. The affine Takagi-Sugeno (TS) fuzzy model consists of rules R_i with the following structure:

$$R_i: \text{If } x \text{ is } A_i \text{ then } y_i = a_i^T x + b \quad i = 1, 2, \ldots, K \qquad (1)$$

antecedent consequent

where $x \in X \subset R^p$ is a crisp input vector, A_i is a (multidimensional) fuzzy set: $\mu_{Ai}(x)$: $X \rightarrow [0, 1]$, $y_i \in R$ is the scalar output of the i-th rule, $a_i \in R^p$ is a parameter vector

Corresponding Author: L. Mehennaoui, Department of Electrotechnic, Skikda University, Algeria

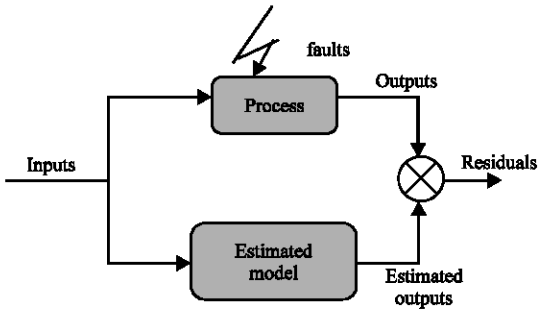


Fig. 1: General procedure of residual generation

and b_i is a scalar offset. K is the number of rules in the rule base.

Given the outputs of the individual consequents y_i, the global output y of the TS model (1) is computed using the weighted (fuzzy) mean formula

$$y = \sum_{i=1}^{K} \beta_i(x) y_i / \sum_{i=1}^{k} \beta_i(x) \qquad (2)$$

Here $\beta_i(x)$ denotes the degree of fulfilment of the i-th rule's antecedent, $\beta_i = \mu_{Ai}(x)$.

For building fuzzy models from data, generated by poorly understood dynamic systems, the input-output representation is often applied. The most common structure is the NARX (Nonlinear AutoRegressive with eXogenous input) model.

In terms of rules, the model is given by

R_i: If y(k) is $A_{i,1}$ and y(k-1) is $A_{i,2}$ ... and y(k-n_y+1) is $A_{i,ny}$ and u(k) is $B_{i,1}$ and u(k-1) is $B_{i,2}$ and ... and u(k-n_u+1) is $B_{i,nu}$ then (3)

$$\hat{y}(k+1) = \sum_{j=1}^{n_y} a_{i,j} y(k-j+1) + \sum_{j=1}^{n_u} b_{i,j} u(k-j+1) + c_i$$

where k denotes discrete time sample, n_y and n_u are integers (fixed by the user) related to the system's order and a_i, b_i, c_i are consequent parameters. The NARX model can represent MISO systems directly and MIMO systems in decomposed form of a set coupled MISO models.

By choosing the structure of the model, the identification problem is transformed into static nonlinear regression y=F(x). The model input x is called the regressor, the output y is called the regressand and the product space of the regressor and the regressand, Z=(X xY)$\subset R^n$ is called he regression space, where n=p+1 is the dimension of this space. Recall that p is the dimension of the regressor vector x. In this space, the equation y=F(x) defines a hypersurface (subspace of the dimension R^p), wich is called regression surface. Geometrically, the consequents of the affine TS model (1) can be seen as hyperplanes in the regression space. By means of the antecedent fuzzy sets, the regression space is partitioned into smaller regions, in which the regression surface can be locally approximated by these hyperplanes. The purpose of identification is to find the number, locations and parameters of the hyperplanes, such that the regression surface is accurately approximated. This is achieved by applying a class of fuzzy clustering methods called subspace clustering algorithms. In this study, the Gustafson-Kessel(GK) algorithm's is used.

Gustafson-kessel(GK) algorithm: First, we have to construct a matrix Z, of data to be clustered. This is achieved by concatenating a matrix containing the regressions vectors in its columns and a vector containing the regressands.

As an example, consider a SISO system for wich a set of N measurement is available :

$$S=\{(y(k),u(k)):k=1, 2, \ldots, N\}$$

Postulating, for instance, a second order NARX structure, y(k+1)=F(y(k),y(k-1),u(k),u(k-1)), the data set for clustering is constructed as:

$$Z = \begin{pmatrix} y(2) & y(3) & \ldots & y(N-1) \\ y(1) & y(2) & \ldots & y(N-2) \\ u(2) & u(1) & \ldots & u(N-1) \\ u(1) & u(2) & \ldots & u(N-2) \\ y(3) & y(4) & \ldots & y(N) \end{pmatrix} \qquad (4)$$

The first four rows contain the regrssors and the last row the regressand. The vector in the k-th column of the matrix Z will be denoted by z_k.

The set of vectors z_k, k = 1, 2,..., N will be partitioned into c clusters, represented by their prototypical vectors $v_i = [v_{i,1}, \ldots, v_{i,n}]^T \in R^n$, i=1,...,c.

Denote $V \in R^{n \times c}$ the matrix having v_i in it's column. V is called the prototype matrix. The fuzzy partitioning of the data among the c clusters is represented as the fuzzy partition matrix $U \in R^{cxN}$ whose elements denoted $\mu_{i,k} \in [0,1]$ are the membership degree of the data vector z_k in i-th cluster. A class of clustering algorithms search for the partition matrix and the cluster prototypes such that the following objective function is minimized

$$J(Z;V,U) = \sum_{i=1}^{c} \sum_{k=1}^{N} \left(\mu_{i,k}\right)^m d^2\left(z_k, v_i\right) \qquad (5)$$

subject to the following constraints

$$\sum_{i=1}^{c}\mu_{i,k}=1,k=1,...,N \tag{6}$$

$$0<\sum_{k=1}^{N}\mu_{i,k}<N=1,i=1,...,c \tag{7}$$

In (5), m>1 is a parameter that control the fuzziness of the clusters. The usual setting with m=2 is suitable for most applications. The function $d(z_k, v_i)$ is the distance of the data vector z_k from the cluster prototype v_i. The constraint[18] avoids the trivial solution U=0 and the constraint[10] guarantees that clusters are neither empty nor contain all the points to degree 1.

The optimization problem defined by the functional[9] subject to the constraints[18,10] can be solved by different nonlinear optimization techniques. The most popular one is the so-called fuzzy c-means algorithm[6]. Gustafson and Kessel extended the c-means algorithm for an inner-product metric norm

$$d^2(z_k,v_i)=(z_k-v_i)^T M_i(z_k-v_i) \tag{8}$$

where M_i ia appositive definite matrix adapted according the actual shapes of the individual clusters, described approximately by the cluster covariance matrices F_i

$$F_i=\frac{\sum_{k=1}^{N}(\mu_{i,k})^m(z_k-v_i)(z_k-v_i)^T}{\sum_{k=1}^{N}(\mu_{i,k})^m} \tag{9}$$

The distance inducing matrix M_i is calculated as the normalized inverse of the cluster covariance matrix

$$M_i=\det(F_i)^{\frac{1}{n}}F_i^{-1} \tag{10}$$

In the iterative optimization scheme of the GK algorithm below, the subscript (l) denotes the value of a variable at the l-th iteration.

Gustafson-kessel fuzzy clustering algorithm: Given the data matrix Z, choose the number of clusters $1 < c < N$, the weighting exponent m>1 and the termination tolerance $\varepsilon>0$. Initialize the fuzzy partition matrix $U^{(0)}$ randomly, such that is satisfies the conditions[18,10].
Repeat for l=1,2,…

Step1: Compute the cluster prototypes (means):

$$v_i^l=\frac{\sum_{k=1}^{N}(\mu_{i,k}^{(l-1)})^m z_k}{\sum_{k=1}^{N}(\mu_{i,k}^{(l-1)})^m},1\le i\le c \tag{11}$$

Step2: Compute the cluster covariance matrices

$$F_i=\frac{\sum_{k=1}^{N}(\mu_{i,k}^{(l-1)})^m(z_k-v_i^{(l)})(z_k-v_i^{(l)})^T}{\sum_{k=1}^{N}(\mu_{i,k}^{(l-1)})^m},1\le i\le c \tag{12}$$

Step3: Compute the distances:

$$d_{i,k}^2=(z_k-v_i^{(l)})^T[\det(F_i)^{\frac{1}{n}}F_i^{-1}](z_k-v_i^{(l)}),\quad 1\le i\le c,1\le k\le N \tag{13}$$

Step4: Update the fuzzy partition matrix:

$$\mu_{i,k}^{(l)}=1/\sum_{j=1}^{c}(d_{i,k}/d_{j,k})^{2/(m-1)},1\le i\le c,1\le k\le N \tag{14}$$

if $d_{i,k}=0$ for some i=s, set $\mu_{k,s}=1$ and $\mu_{i,k}=0\forall i\neq s$
until $\|U^{(l)}-U^{(l-1)}\|<\varepsilon$

This algorithm simply loops through the estimates of the cluster centres V, the covariance matrices F and the fuzzy partition matrix U. We explain, now, how to derive fuzzy models from these matrices.

Estimation of consequent parameters: There are several methods to obtain the consequent parameters. Since the model should serve as numerical predictor, we use the global least square approach, which gives the least prediction error.

In order to obtain an optimal global predictor, the aggregation of the rules should be taken into account. When using the fuzzy mean defuzzification[1], which is a convex linear combination, a global least squares problem can be solved to obtain the consequent parameter estimates.

The membership degrees $ß_{i,k} = \mu_{Ai}(x_k)$, representing the degree of fulfilment of the i-th rule of each data point, can be obtained from the fuzzy partition matrix U. Recall that each row of U contains a point-wise definition of the membership function for the data in the product space X x Y. In order to obtain the membership function A_i in the regressor space X, the i-th row of U, denoted $U_{(i)}$ must be projected onto regressor space

$$\beta_{i,k} = \text{proj}_{N_p}^{N_{p+1}}(U_{(i)}) \quad (15)$$

where proj(.) is the point-wise projection operator[ ].

The result of the projection step is that a set of data vectors with repeated regressors x_k are assigned the maximum membership degree from this set. in order to write[1] in a matrix form for all data (x_k, y_k), $1 \le k \le N$, denote B_i a diagonal matrix in R^{NxN} having the normalized membership degree γ_i as its k-th diagonal element. Finally denote X' the matrix in R^{NxcN} composed from matrices produces of B_i and X_e as:

$$X' = [(B_1X_e); (B_2X_e); \ldots; (B_cX_e)]^T \quad (16)$$

Denote θ' the vector in $R^{c(p+1)}$ given by

$$\theta' = [\theta_1^T, \theta_2^T, \ldots, \theta_c^T] \quad (17)$$

where $\theta_i = [a_i^T, b_i^T]$ for $1 \le i \le c$.

The resulting global least square problem $X'[\theta'] \approx y$ has the solution

$$\theta' = [(X')^T X']^{-1}(X')^T y \quad (18)$$

From (17) the parameters a_i and b_i are obtained by

$$a_i = [\theta'_{q+1}, \theta'_{q+2}, \ldots, \theta'_{q+p}], \quad b_i = [\theta_{q+p+1}], q = (i-1)(p-1) \quad (19)$$

Deriving antecedent membership functions: The fuzzy partition matrix u projected onto the antecedent space defines the membership functions point-wise, for the available data. In order to obtain a prediction model, the antecedent membership functions need to be expressed in a form that allows one to compute the membership degrees for any input data. This can be achieved by using an inverse of the distance function of the clustering algorihm in the antecedent product space.

The degrees of fulfilment of the rules are computed by evaluating the distance function, see[11], only for the regressor x and the regressor part of the cluster prototype, using the corresponding partition of the cluster covariance matrix

$$F^x = [f_{ij}], 1 \le i, j \le p \quad (20)$$

The inner product norm then measures the distance of the antecedent vector from the projection of the cluster center to the antecedent space. Then the inner product norm can be evaluated as

$$d(x_k, v^x_i) = (x\text{-}v^x i)T\ F^x_i\ (x\text{-}v^x_i) \quad (21)$$

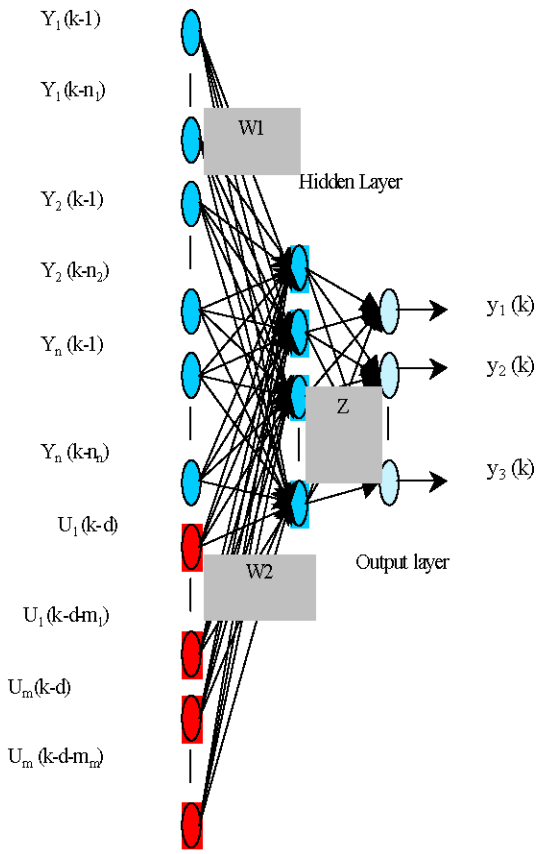


Fig. 2: Two-layer neural network

and transformed into the membership degree (degree of fulfilment), using some kind of inversion. One possible choice is to use the same formula as in the clustering algorithm

$$\beta_i(x_k) = \frac{1}{\sum_{j=1}^{c}[d(x_k, v_i^x)/d(x_k, v_j^x)]^{2/(m-1)}} \quad (22)$$

which takes into account all the rules and computes the degree of fulfilment of one rule relative to the other rules. the sum of the membership degrees also equals one as with clustering, hence $\gamma_i = \beta_i$

Summary of the identification procedure: The identification procedure can be summarized in the following steps:

- Step1: Design identification experiments and collect a set of representative measurements.
- Step2: Choose the model structure, see[2]

- Step3: Cluster data by using GK algorithm.
- Step4: Generate the rules by computing the consequent parameters and the antecedent membership functions.
- Step5: Validate the model.

Residual generation by neural network: It is relevant to use the high potential of neural networks for nonlinear system modelling in the context of fault diagnosis of nonlinear dynamic systems. The most commonly used neural network architecture is the Multilayer Perceptron (MLP) network[17].

Its implementation goes through the following steps:

- Off-line construction of a database using expert knowledge of the process characteristics under different operating conditions.
- Selection of the neural network structure: the NNARX model is recommended[10,17] when the system under consideration is deterministic or weakly noisy. The NNARX model may be represented by the general form:

$$\hat{y}(k) = g(\varphi(k)) \tag{23}$$

the regression vector and the nonlinear function $\varphi^T(k) = (y(k-1)...y(k-n),u(k-d)...u(k-d-m))$ is the regression vector and the nonlinear function g can be realized by a suitable MLP network.

A multivariable NNARX model can be adequately implemented as a feedforward two-layer perceptron network having one hidden layer and an output layer as shown in Fig. 2.

The vector φ(k) of delayed outputs and inputs of the system is applied to the network inputs. ($n_1,\ldots,n_n$, $m_1,\ldots,m_m$,d) are the structural indices also referred to as the lag space of the neural model. The input delay d is generally taken as one.

The hidden layer includes a sufficient number nh of sigmoid units (nh must be specified experimentally) and the output layer contains linear units.

W = (W1 W2) is the weight matrix relating the inputs to the hidden layer units and Z is the weight matrix relating the hidden layer units to the output units.

The neural network outputs are given by:

$$\hat{y}(k) = \psi_i\left(\sum_{j=1}^{n_h} Z_{ij} h_j(k) + z_{i0}\right) \quad I=1,\ldots,n \tag{24}$$

$$h_j(k) = \phi_j\left(\sum_{l=1}^{n_\varphi} W_{jl}\varphi_l(k) + w_{j0}\right) j = 1,..,n_h \tag{25}$$

where φj are sigmoïd type activation functions and ψ_i are linear type activation function and (w_{j0}, z_{i0}) are the biases.

Network training: The network weights and biases (randomly initialized) are adjusted using a suitable minimisation algorithm of the following mean square error criterion:

$$E_N = \frac{1}{N}\sum_{k=1}^{N}(y(k)-\hat{y}(k))^T(y(k)-\hat{y}(k)) \tag{26}$$

where N is the length of the training data set. The Levenberg-Marquardt algorithm is recommended to use as pointed out in[7].

Network validation: In this stage the resulting neural model is evaluated to decide for its adequate representation of the system. This is done by testing the trained network using a data set different from the one used for training. If the trained network is judged unsatisfactory after the validation tests then it is necessary to go backwards in the procedure by retraining the network with different weight initializations, or by generating additional training data, or by modifying the network structure (by redefining the regression vector and the number of hidden units).

As in the case of residual generation by fuzzy sets, all these steps are accomplished off-line. When the neural network is validated, it may be utilized for online residual generation.

Residual evaluation: The task of residual evaluation can be achieved by a fuzzy neural decision scheme[8,9] as represented in Fig. 3.

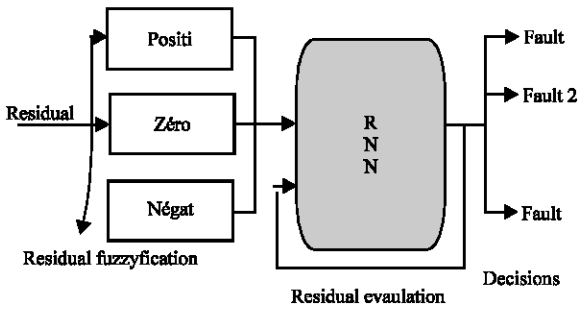


Fig. 3: Neural fuzzy decision scheme

A fuzzy neural network is based on the association of fuzzy logic inference and the learning ability of neural networks. The fuzzy neural approach is a powerful tool for solving important problems encountered in the design of fuzzy systems such as: determining and learning membership functions, determining fuzzy rules, adapting to the system environment. The main points of the residual evaluation procedure are described below:

Residual fuzzyfication: It consists in converting the numerical values of residuals into linguistic variables. Each input (residual) may be described by three linguistic variables (Negative, Zero, Positive). Each linguistic variable is represented by a membership function, which has generally a triangular or trapezoidal shape. The linguistic variable Zero defines the range where the residual may be considered to be unaffected by a fault. The linguistic variables Negative and Positive define the residual amplitude ranges indicating the presence of a fault. The corresponding membership functions give the extent to which a residual is or is not affected by a fault.

Neural network structure: For fault diagnosis, it is desirable to use a neural network to model the nonlinear relationship between the fuzzyfied residuals and the fault decision functions. A multilayer perceptron network is therefore a good candidate. Moreover, to account for memory in the decision process it is necessary to use a Recurrent Neural Network (RNN). The RNN may be implemented as a NNARX model described by:

$$D_k(f_i) = g_i(\varphi(k)) \tag{27}$$

$D_k(f_i)$ i=1…n_f, are the fault decision functions also referred to as fault indicators and fi are the faults acting on the process. The regression vector φ(k) contains the fuzzy residuals Rj(k) j=1..n_r and the delayed decisions $D_{k-1}(f_i)$ i=1…nf. Because of the feedback introduced, the recurrent NNARX model may be realized by a three-layer MLP. This is illustrated by the example given in Fig. 4, which shows a residual evaluation scheme processing three residuals (r1, r2, r3) to diagnose three faults (f1, f2, f3).

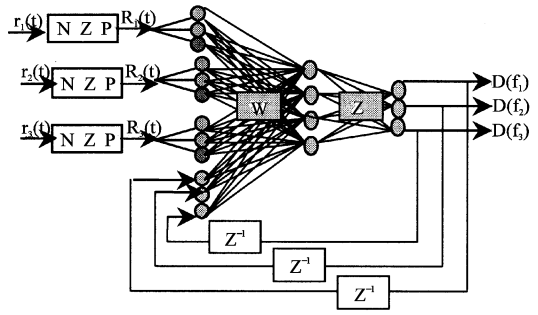


Fig. 4: Example of RNN used for residual evaluation

The corresponding neural network has the following architecture: an input layer with 12 units representing all possible states of the fuzzy residuals together with the past decisions, a hidden layer having 4 units and an output layer with 3 units each assigned to a decision function. The use of this RNN architecture ensures reliable dynamic decision-making[8-10].

Training: Prior to on-line use, network training is performed for all possible fault scenarios. During training a residual pattern corresponding e.g. to fault f_i is applied to the network input and a one is assigned to the corresponding output. The network weights are then adjusted by an appropriate algorithm thus enabling the neural network to learn the imposed input-output pattern. The use of the back propagation algorithm is recommended[11]. The ultimate goal of the training is to achieve the extraction and selection of the necessary parameters defining the «if-then» inference rules.

RESULTS

Simulation results are next presented to assess the capacity of this diagnosis approach based on neural and fuzzy techniques to detect and isolate sensor faults in a nonlinear process. The nonlinear process considered here is composed of three identical tanks having section Q, connected in series by a pipe of section q, with outlet at height H. The system outputs are the three tank levels $y_i = h_i$ i=1…3 satisfying the condition $h_1>h_2>h_3>H>0$.

This system is governed by the following nonlinear differential Eq:

$$\begin{cases} \dot{h}_1 = -b\cdot\sqrt{h_1 - h_2} + (1/Q)\cdot u \\ \dot{h}_2 = b\cdot\sqrt{h_1 - h_2} - b\cdot\sqrt{h_2 - h_3} \\ \dot{h}_3 = b\cdot\sqrt{h_2 - h_3} - b\cdot\sqrt{h_3 - H} \end{cases} \tag{28}$$

$b=q/Q.\sqrt{2g}$, q= 0.196 m^2, Q= 78.54 m^2, H=3 m.

g is the gravity constant and u=1.222m3/sec is the constant input flow. This simulation study is carried out with a sampling time Ts = 10 sec and with initial conditions: h_{10}= 6.9 m, h_{20}= 5.5 m , h_{30}= 4.3 m.

Method using fuzzy sets

Residual generation: The structure of the fuzzy model is selected by using the insight in the physical structure of the system as follows:

Output 1: n_{y11} = 1, n_{y12}= 1, n_{y13} =0, u_{11}=1
Output 2: n_{y21} = 1, n_{y22}= 1, n_{y23} =1, u_{12}=0
Output 3: n_{y31} = 0, n_{y32}= 1, n_{y33} =1, u_{13}=0

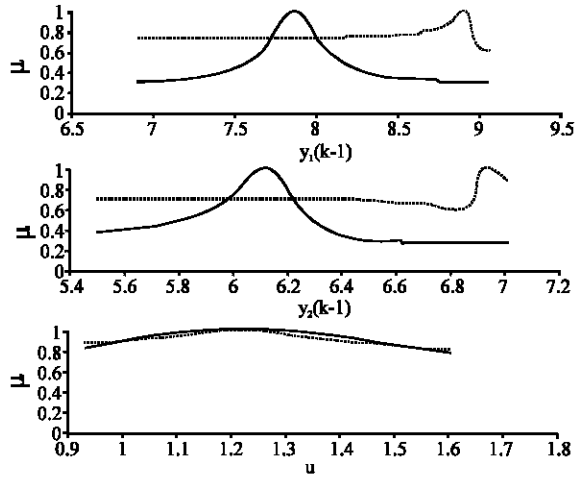


Fig. 5: Membership functions (output1)

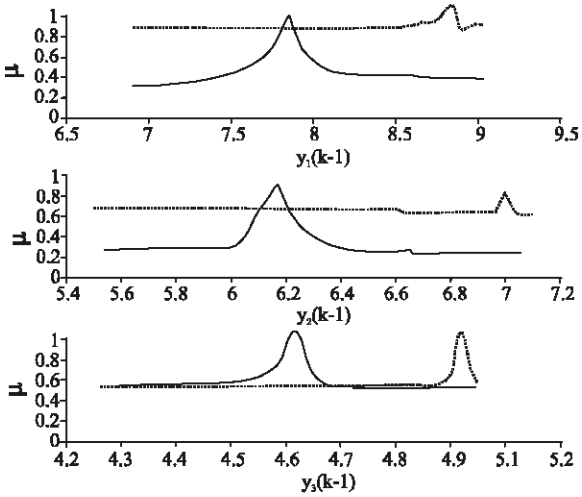


Fig. 6: Membership functions (output2)

As example, the degrees selected for the output 1 state that the level h_i depends on h1, h2, u, but not on h3.

The number of clusters is c=2, then the number of rules are also 2. The fuzzy TS models obtained are:

Output 1:

R_1: If $y_1(k-1)$ Is A_{11} And $y_2(k-1)$ Is A_{12} And u Is A_{13}
Then $y_1(k) = 0.96\, y_1(k-1)+0.05\, y_2(k-1)+0.13u -0.07$

R_2: If $y_1(k-1)$ Is A_{21} And $y_2(k-1)$ Is A_{22} And u Is A_{23}
Then $y1(k) = 0.97\, y_1(k-1)+0.04\, y_2(k-1)+ 0.12\, u- 0.08$

The cluster centers are regrouped in the Table 1.

The antecedent membership functions obtained are represented by the Fig. 5.

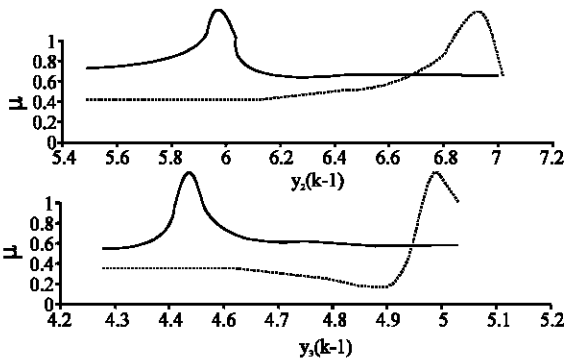


Fig. 7: Membership functions (output3)

Output 2:

R_1: If $y_1(k-1)$ Is A_{11} Andy$_2(k-1)$ Is A_{12}And $y_3(k-1)$ Is A_{13} Then $y_2(k) = 0.035\, y_1(k-1) +0.922\, y_2(k-1) + 0.05\, y_3(k-1) - 0.002$

R_2:. If $y_1(k-1)$ Is A_{21} And $y_2(k-1)$ Is A_{22}And $y_3(k-1)$ Is A_{23} Then $y_2(k) = 0.037\, y_1(k-1)+0.926\, y_2(k-1)+ 0.036\, y_3(k-1) +0.006$

Output 3:

R_1: If $y_2(k-1)$Is A_{11}And $y_3(k-1)$Is A_{12}
Then $y_3(k) = 0.05\, y_2(k-1)+0.907\, y_3(k-1)+0.144$

R_2: If $y_2(k-1)$ Is A_{21} And $y_3(k-1)$ Is A_{22}
Then $y_3(k) = 0.04\, y_2(k-1)+0.920\, y_3(k-1)+0.121$

After validation, this NNARX fuzzy model is used to generate the residuals: $r_i(k)=y_i(k)-\hat{y}_i(k)$ i=1…3. In normal operation, the residuals are near zero as shown in Fig. 8.

Residual evaluation. The linguistic variables describing the fuzzyfied residuals are defined by the following membership functions (MF):

- N : negative residual with trapezoidal MF»,
- Z : zero residual with triangular MF»,
- P : positive residual with trapezoidal MF ».

The membership functions for each residual are given below:

- Résidu 1: N_1 =[-0.5 -0.4 -1e-3 -8e-4],
  Z_1 = [-1e-3 0 1e-3] , P_1 = [0.5e-3 5e-3 2.5 2.5]
- Résidu 2: N_2 = [-1 -0.6 -9e-3 -3e-3],

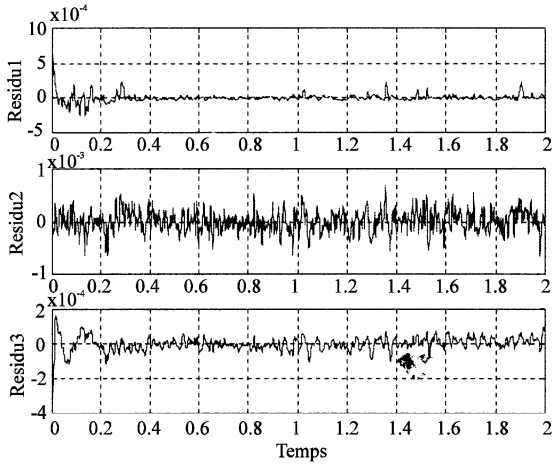


Fig. 8: Residuals by fuzzy sets (normal operation)

Z_2 = [-5e-3 0 5e-3], P_2 = [1e-3 4e-3 1 1]

- Résidu 3: N_3= [-1 -0.5 -1e-3 -6e-4], Z_3 = [-8e-4 0 1e-2], P_3 = [.5e-2 1e-2 1.5 2]

Table 1: Cluster centers (output 1)

rule	y_1(k-1)	y_2(k-1)	u
1	7.87	6.12	1.23
2	8.91	6.93	1.22

Antecedent membership functions

Table 2: Cluster centers (output 2)

rule	y_1(k-1)	y_2(k-1)	y_3(k-1)
1	8.18	6.36	4.66
2	8.91	6.94	4.97

Antecedent membership functions

Table 3: Cluster centers (output 3)

rule	y_2(k-1)	y_3(k-1
1	5.99	4.46
2	6.92	4.96

The RNN used in this simulation study is shown in Fig. 4. Its training is based on the rules summarized in Table 1, which have been obtained after many simulation tests.

The learning operation realized by the back propagation algorithm converged after 3266 iterations with a mean square error E=0.001.

Sensor fault diagnosis of the three-tank process: Various simulation tests have been performed in order to validate the efficiency of this diagnosis scheme and the results are quite conclusive. For illustrative purposes only two fault scenarios summarized in Table 2 and 3 are discussed.

Case 1: Bias type faults are injected in sensors 1 and 2 as described in Table 5. The corresponding residuals are shown in Fig. 9.

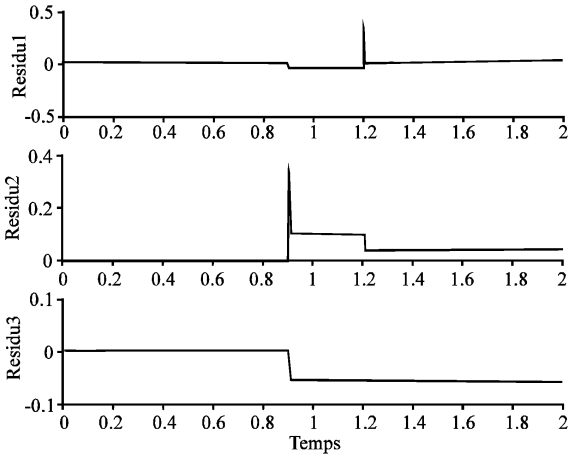


Fig. 9: Residuals by fuzzy sets (case1)

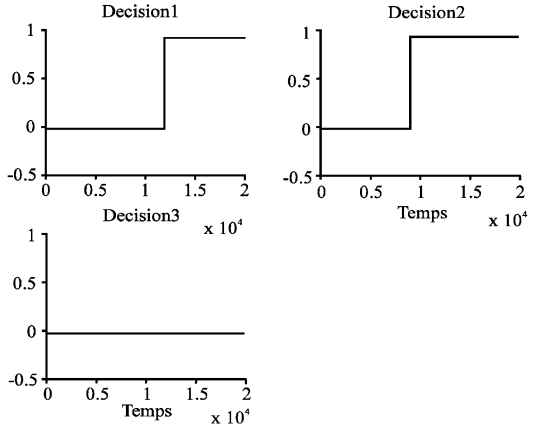


Fig. 10: Decision functions (case1)

The fault f_1 on sensor 1 affects positively the residual r_1 and negatively the residuals r_2 and r_3 at time t=12000s,

whereas the fault f2 on sensor 2 affects positively the residual r_2 and negatively the residuals r_1 and r_3 at time 9000s.

The obtained decision functions allow to well detect the faults f_1 and f_2 as shown in figure 10. It was possible by use of fuzzyfied residuals and the training network operation.

Case 2: This fault scenario in sensors 2 and 3 is described in Table 6. The corresponding residuals are shown in Fig. 11.

The fault f_2 on sensor 2 affects positively the residual r_2 and negatively the residuals r_1 and r_3 at time t=13000, whereas the fault f_3 on sensor 3 affects positively the residual r_3 and negatively the residuals r_1 and r_2 at time t=10000.

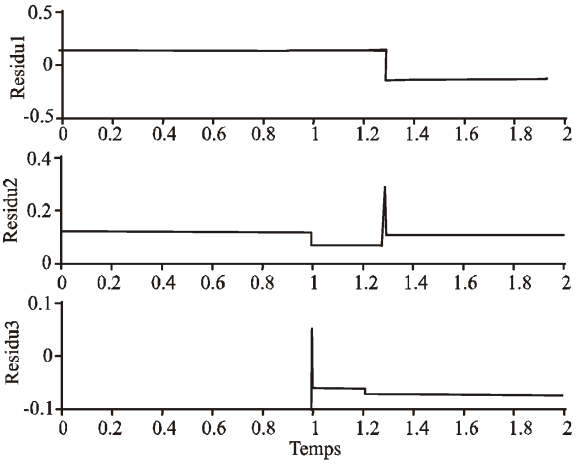


Fig. 11: Residuals by fuzzy sets (case2)

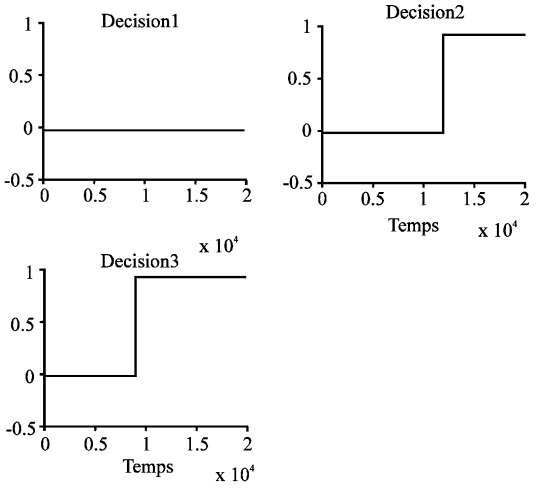


Fig. 12: Decision functions (case2)

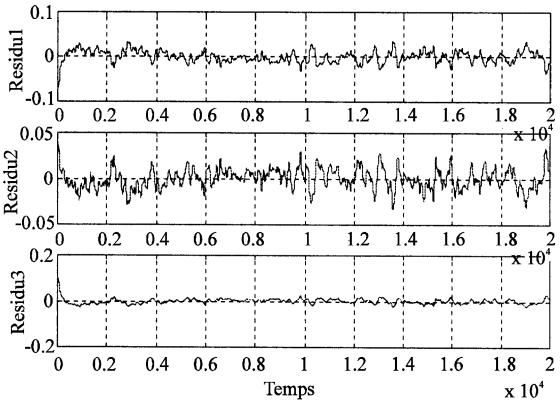
As shown in Fig. 12, the fault indicators detect and isolate successfully the faulty sensors.

METHOD USING NEURAL NETWORK

Residual generation by neural network: A NNARX model having the architecture shown in Fig. 2 has been used with the following parameters:
$n_1=n_2=n_3=m_1=1$, d=1, $n_f=4$, $n_h=4$. Training of this network was done by the Levenberg-Marquardt algorithm and the

Table 4: Inference table

N°	N1	Z1	P1	N2	Z2	P2	N3	Z3	P3	D1	D2	D3
1	0	1	0	0	1	0	0	1	0	0	0	0
2	0	0	1	1	0	0	1	0	0	1	0	0
3	1	0	0	0	0	1	1	0	0	0	1	0
4	1	0	0	1	0	0	0	0	1	0	0	1
5	0	0	1	1	0	0	0	0	1	1	0	1
6	1	0	0	0	0	1	0	0	1	0	1	1
7	0	0	1	0	0	1	1	0	0	1	1	0



13: Residuals by neural network in normal operation

Table 5: case1

Sensor N°	Fault time	Fault magnitude
1	12000	0.7
2	9000	0.6

Table 6: case 2

Sensor N°	Fault time	Fault magnitude
2	13000	0.3
3	10000	0.5

mean square error reached at 500 iterations is $E= 2.36510^{-4}$. After validation, this NNARX model is used to generate the residuals:
$r_i(k)=y_i(k)-\hat{y}_i(k)$ i=1…3. In normal operation, the residuals are near zero as shown in Fig. 13.

Residual evaluation: The linguistic variables describing the fuzzyfied residuals. In this case, the membership functions are given as follow:

- Résidu 1: N1 = [-1 -1 -0.09 -0.085] ,
  Z1 = [-0.15 0 0.15] , P1 = [0.05 0.06 1 1]
- Résidu 2 : N2 = [-1 -1 -0.05 -0.040],
  Z2 = [-0.08 0 0.08], P2 = [0.045 0.055 1 1]
- Résidu 3 : N3= [-1 -1 -0.04 -0.030],
  Z3 = [-0.08 0 0.25], P3 = [0.15 0.2 1 1]

We use the same RNN shown in Fig. 4. Its training is based on the rules summarized in Table 4. We notice that is the same logic decision for both methods.

Sensor fault diagnosis of the three-tank process: The same scenario of faults.

Case 1: Bias type faults are injected in sensors 1 and 2 as described in Table 2. The corresponding residuals are shown in Fig. 14.

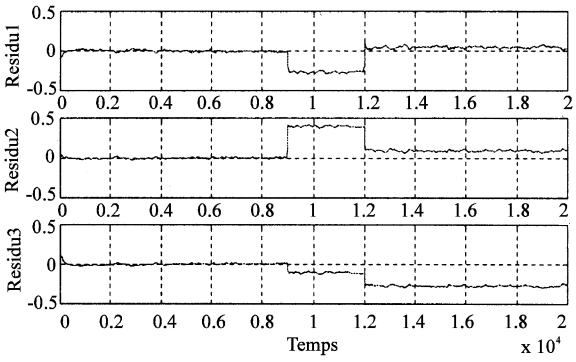


Fig. 14: Residuals by neural network (case1)

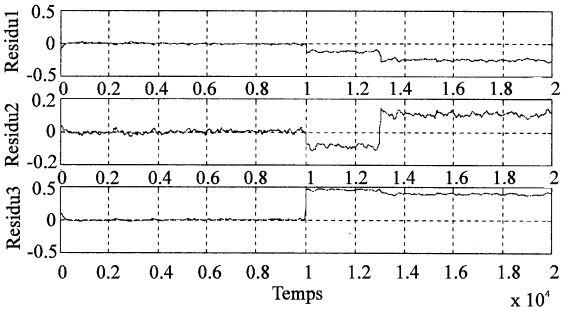


Fig. 15: Residuals by neural network (case2)

We notice that effects of faults on this residuals are similar with those on the residuals obtained by the method of fuzzy sets.
also with this method, the decision functions isolate the two faults and we obtain the same function decision shown in Fig. 7.

Case 2: This fault scenario is the same as that described in Table 3. The corresponding residuals are shown in Fig. 15.

With this method, the faulty sensors are also isolated successfully and we obtain the same decision functions shown in Fig. 12.

CONCLUSION

A fuzzy neural scheme for on-line fault diagnosis was presented. A NNARX model is used for residual generation. This NNARX model can be obtained either by fuzzy sets or neural network. A recurrent fuzzy neural network performs the residual evaluation task. Fault diagnosis is achieved by training the network to recognize the pattern of the fault signatures. Preliminary simulation results show the efficiency of the developed scheme for detecting and isolating sensor faults in a nonlinear system. The applicability of this qualitative diagnostic approach to the case of system actuator and component faults is currently under study.

REFERENCES

1. Benloucif, M.L. and L. Mehennaoui, 1992. Fault sensor detection and isolation in automated systems. Proceedings of the International Seminar of Signal and Automatic Systems, SSA, Blida.
2. Benloucif, M.L., A. Cheddadi and A. Bouhenchir, 1998. Design of a structured residual generator for fault detection and isolation. Proceedings of the International Conference ICEL, Oran.
3. Frank, P.M., 1990. Fault diagnosis in dynamic systems using analytical and knowledge based redundancy, A survey and some new results, Automatica, 26: 459-474.
4. Patton, R. and J. Chen, 1997. Observer-based fault detection and isolation: Robustness and applications," Control Eng. Practice, 5: 671-682.
5. Benloucif, M.L. and M. Staroswiecki, 2002. Fault diagnosis using a robust estimation method. Proceedings of the International Conference CIFA-Nantes, France.
6. Garcia, E. and P. Frank, 1997. Deterministic nonlinear observer based approaches to fault diagnosis: A survey. Control Eng. Practice, 5: 663-670.
7. Jiang, B., M. Staroswiecki and V. Cocquempot, 2001. Robust observer based fault diagnosis for a class of nonlinear systems with uncertainty. Proceedings of the 40th IEEE Conference on Decision and Control, CDC'01, Orlando.
8. Alexandru, M., C. Combastel and S. Gentil, 2000. Diagnostic Decision using Recurrent Neural Networks. Proceedings of the IFAC Safeprocess ' Budapest, Hungary.
9. Benloucif, M.L. and L. Mehennaoui, 2002. A mixed analytical neuro-fuzzy approach for fault diagnosis. Proceedings of the 2ND Instrumentation and Measurement in Petroleum Applications Conference IMPAC-2002, Boumerdès.
10. Chen, Y.M. and M.L. Lee 2002. Neural networks based scheme for system failure detection and diagnosis. Mathematics and Computers in Simulation, 58: 101-109.
11. Evsukoff, A., C. Combastel and S. Gentil, 1999. Qualitative reasoning and neural network decision procedures for fault detection and isolation. Proceedings of the 14th IFAC World Congress, Beijing, China.
12. Frank, P.M., 1994. Application of fuzzy logic to process supervision and fault diagnosis, Proc. IFAC Safeprocess, Espoo, Finland.

13. Isermann, R., 1998. On fuzzy logic applications for automatic control, supervision and fault diagnosis. IEEE Trans. on Systems, Man and Cybernetics, pp: 28.
14. Schneider, H. and P.M. Frank, 1996. Observer based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation. IEEE Trans. On Control System Tech., 4: 274-282.
15. Theilliol, D., D. Sauter and L.G. Vela valdes, 1997. Integration of qualitative and quantitative methods for fault detection and isolation. Proceedings of the IFAC Safeprocess, Hull, U.K.
16. Hellendoorn, H. and D. Driankov, 1997. Fuzzy Model Identification, Selected approaches. Springer Verlag, London.
17. Norgaard, M., O. Ravn, N.K. Poulsen and L.K. Hansen, 2000. Neural network for modelling and control of dynamic systems. Springer Verlag, London.
18. Bezdec, J.C., R. Hathaway, R.E. Howard, C.A. Wilson and M.P.Windham, 1987. Local convergence analysis of a grouped varaiable version of coordinate descent. J. Optimization Theory and Applications, 54: 471-477.