COMPONENT FAULT DIAGNOSIS USING CO-ACTIVE NEURO-FUZZY SYSTEMS

Letitia Mirea, Lavinia Ferariu

"Gh. Asachi" Technical University of Iaşi
Department of Automatic Control
Blvd. D. Mangeron 53A, RO-700050 Iaşi, Romania
Fax: +40-232-214290, E-mail:{ lmirea, lferaru}@ac.tuiasi.ro

Abstract: The paper investigates the development of the a type of a Co-Active Neuro-Fuzzy System (CANFS) and its application to Fault Detection and Isolation (FDI). Hybrid learning, based on a fuzzy clustering algorithm and a gradient-like method, is used to train the CANFS. The experimental case study refers to the component fault diagnosis of a Three-Tank System. A neuro-fuzzy simplified observer scheme is used to generate the residuals (symptoms) in the form of the one-step-ahead prediction errors. These are further analysed by a neural classifier in order to take the appropriate decision regarding the behaviour of the considered process.

Keywords: fault diagnosis, neuro-fuzzy systems, fuzzy clustering, neural networks, neural classifier

1. INTRODUCTION

During the 1990s, fault diagnosis have become an important field of automatic control with a goal focussed on the detection and isolation of process faults or malfunctions and the diagnosis of undesirable system behaviour. Usually, a fault manifests as a deviation of at least one characteristic property or variable of the process with respect to the corresponding nominal values.

A fault diagnosis system should satisfy requirements such as early detection of small faults with abrupt or incipient time behaviour, diagnosis of faults in different parts of the investigated (supervised) process, detection of faults in closed-loop and supervision of the process in the transient states. All these requirements should be satisfied in the face of the existing measurement uncertainty, disturbances and incomplete knowledge about the process. An efficient fault diagnosis system should avoid interpreting the unknown inputs (disturbances, measurement noise, modelling errors) as faults, i.e. false alarms.

Process control and supervision often require accurate process models. Most processes are non-linear and, therefore, their model should be non-linear (Patton *et al*., 2000). Neural networks have been shown to possess good non-linear function approximation capabilities and have been used in non-linear process modelling. They can also cope with the roubustness problem in FDI. However, the neural model obtained is considered to be a "black-box" model since it is difficult to interpret.

Within a specific operating region, a linear model can approximate the non-linear process behaviour with a reasonable accuracy. Takagi and Sugeno (1985) used a fuzzy modelling approach in which each model input is assigned with several fuzzy sets characterised by a membership function. Through logical combination of these fuzzy inputs, the model-input space is partitioned into several fuzzy regions. A locally linear model is used within each region. The global model output is obtained through the weighted average of the local model outputs.

In the Tagaki-Sugeno fuzzy model, the local linear models can be replaced by neural networks. In this way, a non-linear neural model is developed in each fuzzy partition of the input space. For this purpose, the Functional-Link Neural Networks (FLNNs) are used. This new structure is implemented by a Co-Active Neuro-Fuzzy System (CANFS) that combines the capability of fuzzy reasoning in handling uncertain information and the capability of neural networks in learning from examples (Jang, 1995).

In order to be used to model a non-linear dynamic system, the CANFS should be equipped with dynamic elements. An approach is to use external delay elements.

Process input-output data are used to train the

CANFS, i.e. to determine the parameters that would minimise a performance index. Firstly, a fuzzy clustering algorithm is used to determine the number of fuzzy operating regions and the initial values for the membership functions. Then, gradient-based learning algorithm is applied in order to refine the parameters of the membership functions and to determine the parameters of the local neural models (Jang, 1995; Zhang and Morris, 1996).

The paper is organised in six sections as follows. In Section 2, the principles of fuzzy and neuro-fuzzy modelling are presented. The architecture and the learning procedure for CANFS are presented in Section 3. Section 4 refers to the design of an FDI system based on CANFS (residual generation) and neural networks (residual evaluation). The application of the CANFS to the component fault diagnosis of a laboratory setup (the Amira's Three-Tank System) is presented in Section 5. The conclusions are given in Section 6.

2. FUZZY MODELLING

The Fuzzy Inference System (FIS) is a framework based on the concepts of fuzzy sets, fuzzy rules and fuzzy reasoning. It has been successfully applied in fields such as automatic control, data classification, decision analysis and computer vision. The basic structure of a FIS consists of three main components: (1) a rule base which contains a selection of fuzzy rules, (2) a database which defines the membership functions used in fuzzy rules and (3) a reasoning mechanism which performs the inference upon the rules and a given condition to derive a reasonable conclusion (output).

One of the most applied FIS structures is the Sugeno fuzzy model proposed by Takagi and Sugeno in (Takagi and Sugeno, 1985). A typical fuzzy rule in a Takagi - Sugeno fuzzy model has the form:

Rule i:

if x_1 is A_1 ***and*** x_2 is A_2 ***and*** ... ***and*** x_n is A_n ***then*** $z_i = f(x_1, x_2, ..., x_n)$,

where A_j, $j = 1,...,n$ are fuzzy sets in the antecedent part of i-th rule, while $z_i = f(x_1, x_2, ..., x_n)$ is a crisp function in the consequent part of i-th rule. Usually, $f(x_1, x_2, ..., x_n)$ is a polynomial in the input variables x_j, $j = 1,...,n$, but it can be any function.

Each fuzzy rule can be interpreted within a local modelling framework. The consequence function $f(x_1, x_2, ..., x_n)$ of each rule can be considered to constitute a local model, defined by a set of parameters. The antecedent part of each rule, defined by the fuzzy sets: A_j, $j = 1,...,n$, determines the regime of each local model or a subset of the input space over which this local model applies. The rule firing strengths defined by:

$$w_i = \prod_{j=1}^{n} A_j , \qquad (1)$$

give the validity function of each local model. Since each rule has a crisp output, the overall output is obtained via weighted average:

$$z = \left(\sum_{i=1}^{M} w_i z_i \right) / \left(\sum_{i=1}^{M} w_i \right), \qquad (2)$$

where w_i are the firing strengths of i-th rule (Jang, 1995) and M is the number of fuzzy rules.

A fuzzy model with neural local models can be implemented by a special type of neural network, namely the Co-Active Neuro-Fuzzy System (CANFS) (Jang, 1995). In this approach, function $f(x_1, x_2, ..., x_n)$ is implemented by a functional-link neural network.

The identification of dynamic systems requires models with adequate memory. For this reason, the CANFSs have to be provided with dynamic elements and appropriate learning methods (Mirea and Marcu, 2002). An approach approach refers to CANFS with external dynamics (Jang, 1995; Zhang and Morris, 1996), i.e. static CANFSs provided with external cascades of filters.

3. CO-ACTIVE NEURO-FUZZY SYSTEMS

The proposed CANFS architecture is presented in Fig. 1. In contrast with the Takagi-Sugeno fuzzy modelling approach, in this case each local model is described by a neural network. Functional-Link Neural Networks (FLNNs) are considered.

The FLNN has been developed as an alternative architecture to the multi-layer perceptron network with application to function approximation and pattern recognition. The FLNN is a feed-forward single layer neural network with a number of enhancement nodes referred to as functional links. These are used as supplementary inputs within the network (Pao, *et al.*, 1994). In the following, the functional expansion given by a sub-set of orthogonal trigonometric functions is considered. This provides a more compact representation of the function to be approximated, in the mean-square sense, than other orthogonal basis functions (Patra, *et al.*, 1999).

The generic structure of an FLNN is depicted in Fig.2, where the initial inputs of the net u_n, n=1,...,P, are functionally expanded to constitute the actual inputs of the non-linear neuron, v_m, m=1,...,P+R. One considers in the following the functional expansions given by a sub-set of orthogonal trigonometric basis functions and the output node with a hyperbolic-tangent non-linearity. This choice is based on the analysis and results presented in (Chen and Wan, 1999; Pao *et al.*, 1994, Patra and Pal, 1995; Patra *et al.*, 1999).

For a pre-specified order S of the functional expansion, the actual inputs v_m of the neuron are given by the following set:

$$\{u_n, \{\cos(s \cdot \pi u_n), \sin(s \cdot \pi u_n\}_{s=1,...,S}\}_{n=1,...,P} .$$

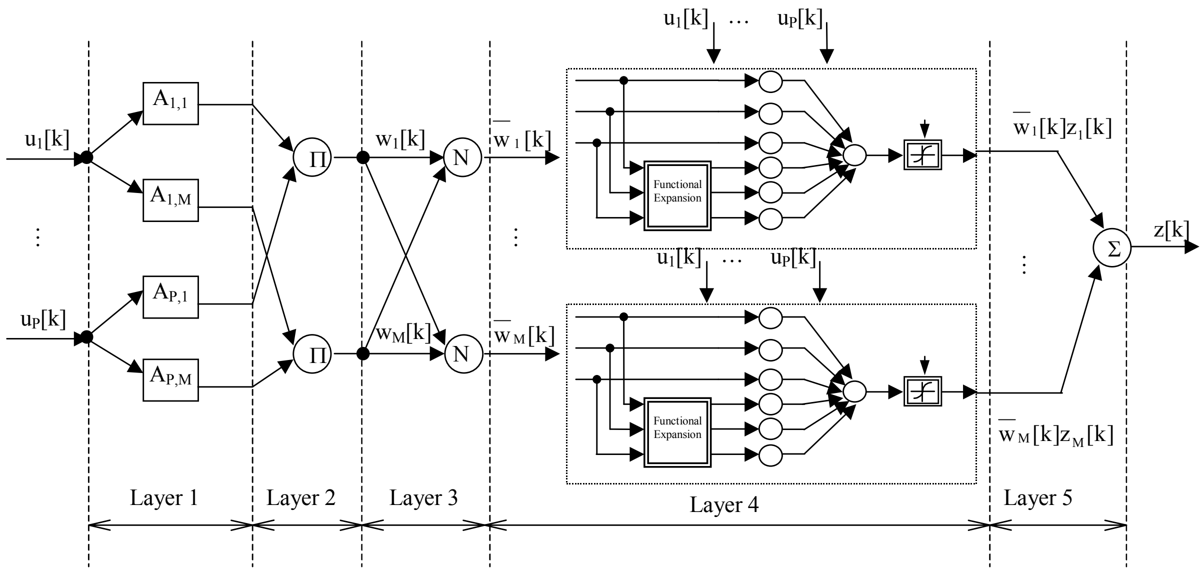


Fig. 1 The architecture of the considered CANFS

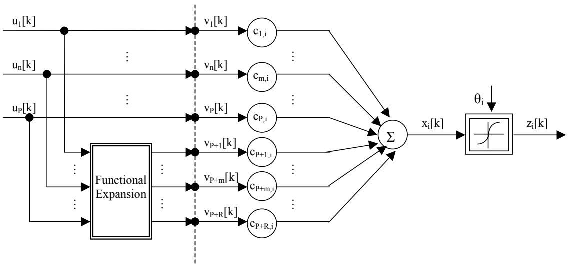


Fig.2 The structure of a Functional-Link Neural Network

Thus, the local model output is given by:

$$z_i[k] = \tanh(x_i[k] + \theta_i), \quad x_i[k] = \sum_{m=1}^{P+R} c_{m,i} \cdot v_m[k]$$
$$\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z}) \qquad ,(3)$$
$$i = 1, \ldots, M$$

where M is the number of the fuzzy rules.

Every node in the <u>1st layer</u> is an adaptive node with the output defined by $\mu_{A_{p,i}}(u_p)$, i=1,...,M, p=1,...,P, where u_p is the input to the node and $A_{p,i}$, $i = 1, \ldots, M$ are the fuzzy sets associated with this node. The outputs of the first layer represent the membership values of the antecedent part of the rules. The membership functions can be any appropriate parameterised membership function, such as the Gaussian function:

$$\mu(x) = e^{-\left(\frac{x-c}{\sigma}\right)^2} . \qquad (4)$$

In relation (4), c represents the centre of the membership function and σ determines the membership function's width. Parameters in this layer are referred to as *premise parameters*.

The <u>2nd layer</u> consists of fixed nodes, which multiplies the incoming signals:

$$w_i[k] = \prod_{p=1}^{P} \mu_{A_{p,i}}(u_p[k]), \quad i = 1, \ldots, M . \qquad (5)$$

In fact, each node output represents the firing strength of a rule. Instead of the product, any other T-norm operator can be used to perform the fuzzy AND operator.

Every node in the <u>3rd layer</u> is a fixed node that computes the normalised firing strength of the i-th rule:

$$\overline{w}_i[k] = \frac{w_i[k]}{\sum_{j=1}^{M} w_i[k]}, \quad i = 1, \ldots, M . \qquad (6)$$

The <u>4th layer</u> consists of adaptive nodes with the output given by $\overline{w}_i \cdot z_i[k]$, where $\overline{w}_i$ is the output of the third layer and $z_i[k]$ is given by relation (3). The parameters in this layer, $\{a_{i,p}, \theta_i, b_{i,j}, d_{i,l}\}$ will be referred to as *consequent parameters*.

The <u>5th layer</u> has a single fixed node that computes the overall output of CANFS as a summation of all incoming signals:

$$z[k] = \sum_{i=1}^{M} \overline{w}_i \cdot z_i[k] . \qquad (7)$$

As a system is usually monitored using sampled data, a discrete time representation of the process is required. The purpose is to identify neuro-fuzzy models for each system output, i.e. Multi-Input Single-Output (MISO) models (Mirea & Marcu, 2002).

For dynamic system identification, these models require spatial representation of time. This is assessed by feeding the CANFS with current and delayed values of the inputs and the outputs of the process (Mirea & Marcu, 2002). For the sake of simplicity, a Single-Input Single-Output (SISO) dynamic system is considered.

Thus, the input-output model obtained using an CANFS is:

$$\hat{y}[k] = f(u_P[k-d], \ldots, u_P[k-d-k_u], \; y_P[k-1], \ldots, y_P[k-k_y]) \qquad , (8)$$

where u_P denotes the process input, y_P represents the process output, and $\hat{y}$ denotes the approximated output given by the trained CANFS. In relation (8), d denotes the dead time and k_u, k_y represents the dynamic orders of the process.

One considers N data pairs collected from the inputs and outputs of the process. In the training stage, the CANFS parameters, collected in a vector ξ, are adapted in order to minimise a quadratic performance index such as the sum-squared error between the CANFS output, $\hat{y}[k]$, and the considered process output, $y_P[k]$. The objective is to ascertain an optimal parameter set ξ^* of the CANFS that minimises the considered performance index:

$$\xi^* = \arg\min_{\xi}\{\frac{1}{2}\sum_{k=1}^{N}(y_P[k]-\hat{y}[k,\xi])^2\}. \qquad (9)$$

A method to select the number of fuzzy rules and the initial values for the premise parameters, based on the training data, is to use a fuzzy clustering algorithm (Chiu, 1994; Mirea & Marcu, 2002). The purpose of the fuzzy clustering algorithm is to distil natural groupings of the CANFS input data set, producing a concise representation of the system's behaviour. Finally, a number of cluster centres are obtained. For each data point a degree of membership to each cluster is computed (Marcu, 1996; Mirea & Marcu, 2002). Based on these values, the standard deviations of each Gaussian membership function are obtained. The resulting cluster centres and standard deviations are used as initial values for the premise parameters and are found using the following gradient method:

$$\begin{aligned} &\varsigma_{new} = \varsigma_{old} + \Delta\varsigma; \quad \Delta\varsigma = -\eta\cdot\frac{\partial E}{\partial \varsigma}; \\ &E = \frac{1}{2}\cdot\sum_{k=1}^{N}(y_P[k]-\hat{y}[k,\varsigma])^2 \end{aligned} \quad . \qquad (10)$$

Relation (10) is used to adapt the consequent parameters as well. For these parameters, an initialisation with small random values is applied. In relation (10), ς is one of the ANFS-LRS premise or consequent parameters and η is the learning rate.

4. NEURO-FUZZY DESIGN OF FDI SYSTEM

4.1 Residual generation

For the generation of symptoms, the ANFS-LRSs replace the analytical models that describe the process. Instead of a multi-input multi-output structure, an CANFS model for each system output is identified, i.e. a MISO model. As the control system operates in closed-loop, faults tend to be hidden by feedback action. Thus, both inputs and outputs of the process are used as inputs of the CANFS.

The neuro-fuzzy models can be then used in an observer-like arrangement (Marcu *et al.*, 2001). Structured sets of symptoms are generated to enable a unique fault diagnosis. This is based on residual signals that are obtained by subtracting the approximations of an observer scheme from the corresponding process measurements. The Neuro-Fuzzy Simplified Observer Scheme (NF-SOS) is described in the sequel. Its design is based on the use of CANFS introduced previously. It is further applied to the considered case study.

The Neuro-Fuzzy Simplified Observer Scheme (NF-SOS). One considers a process with I inputs $u_{P,i}[k]$, i=1,...,I and O outputs $y_{P,j}[k]$, j=1,...,O, all known at sampling time [k]. The NF-SOS consists of a number of MISO neuro-fuzzy systems with each one driven by all inputs and outputs of the process. Each CANFS estimates one output of the system:

$$\begin{aligned} \hat{y}_j[k] = f_{NF\text{-}SOSj}(&\mathbf{u}_P[k-d],...,\mathbf{u}_P[k-d-k_u], \\ &\mathbf{y}_P[k-1],...,\mathbf{y}_P[k-k_y]); \qquad (11) \\ j=1,...,O& \end{aligned}$$

where $\mathbf{u}_P[k]=[u_{P,i}[k]]_{i=1,...,I}$ is the vector of process inputs and $\mathbf{y}_P[k]=[y_{P,j}[k]]_{j=1,...,O}$ is the vector of process outputs.

The resulting bank of neuro-fuzzy models approximates all outputs of the process. The training of the CANFSs is based on the system data corresponding to its normal behaviour. The following residuals are then generated:

$$\varepsilon_j[k] = y_{P,j}[k]-\hat{y}_j[k]; \quad j=1,...,O \qquad (12)$$

These patterns of change are further used to detect and locate the faults.

4.2 Residual evaluation

The residual evaluation stage is actually a classification task. This means to match each pattern of the residual vector with one of the pre-assigned classes of faulty behaviour, if available, and the fault-free case, respectively (Marcu *et al.*, 2001).

The uncertainty in classification of patterns may arise here from the overlapping nature of various classes. For fault diagnosis this is a realistic assumption, especially when incipient faults have to be detected and isolated. Therefore, a robust decision can be achieved by using a neural network as pattern classifier (Marcu *et al.*, 2001). The static Multi-Layer Perceptron (MLP) with sigmoid neurons is considered here.

The neural classifier maps the patterns (12) from the residual space into a decision space. The patterns belonging to a class are made to cluster around pre-selected points, optimally chosen (Marcu *et al.*, 2001). A fault is detected and isolated if an unknown input pattern is mapped closest to one of the decision space target vectors. That multi-dimensional point corresponds to the associated learned class that reflects a fault.

A fault is only detected if the input pattern is mapped far from all learned classes. For the latter case, that is a new (faulty) situation, only the synthesis of the classifier must be reconsidered for further fault diagnosis. One simple criterion used in the decision logic is based on the minimum Euclidean distance to the target vectors of the classifier.

5. EXPERIMENTAL RESULTS

The methodology presented is assessed by using real data from the Three-Tank System laboratory setup

(Amira, 1993). The study refers to the component fault detection and isolation. The experimental set-up consists of three cylindrical tanks with identical cross sections being filled with water. Circular pipes interconnect the tanks (Fig. 3).

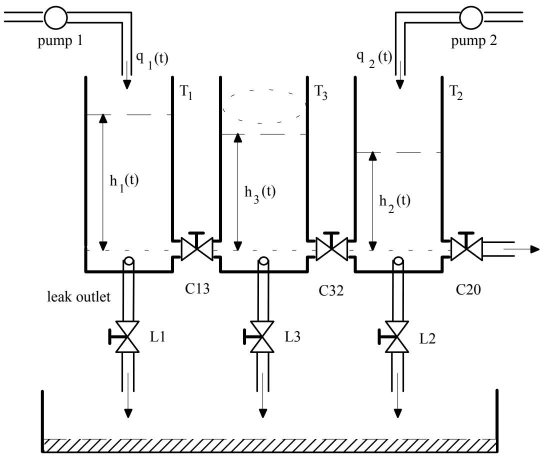


Fig. 3. The Three-Tank System DTS200.

An analytical model of the system is represented by three first-order non-linear differential equations. That model is used by an appropriate strategy to control the water inlet by two pumps. The volume flows $q_1(t)$ and $q_2(t)$ of lateral tanks T_1 and T_2, respectively, are controlled such that the level in the corresponding tanks, $h_1(t)$ and $h_2(t)$, are pre-assigned independently. The level $h_3(t)$, in the middle tank T_3, is uncontrollable. Here *t* stands for the time variable. The control strategy works at a sampling rate of 0.1 seconds. Although the dynamic modelling of the considered system is relatively simple, the resulted non-linear analytical model is a limited approximation (Marcu, *et al*., 1999).

The connecting pipes and tanks are additionally equipped with manually adjustable valves and outlets for the purpose of simulating clogs and leaks. Four classes of process behaviour have been taken into consideration. These are the normal behaviour (one class, [NB]: valves C_{13}, C_{32}, C_{20} are open, outlets L_1, L_3, L_2 are closed) and incipient faults. The latter refers to a leakage in each tank (three classes, [L_1], [L_3], [L_2]). The simulated faults correspond to about 20-25% opening of the involved outlets.

For the experiments, the reference values of the liquid levels in the lateral tanks were changed pulse-wise with different magnitude and duration for each controlled tank. A test period of 400 seconds was considered. The system data were sampled at every 5 seconds, due to the slow nature of the process. Thirty-five experiments were done in a period of a month in order to take into consideration the influence of the plant environment.

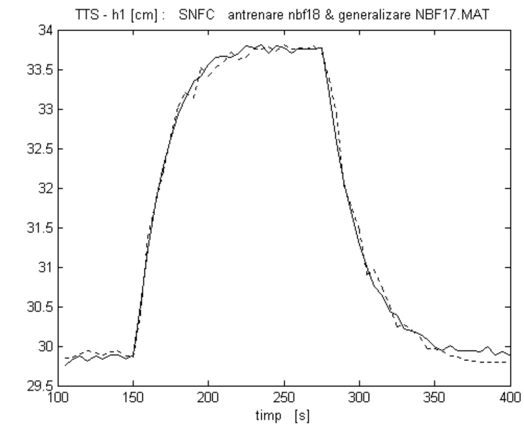
A NF-SOS has been synthesised, based on the CANFSs. The neuro-fuzzy observer has been designed using the real data corresponding to the normal behaviour of process. The purpose is to obtain neuro-fuzzy models for each system output, i.e. the measured liquid levels in each tank:

$$\begin{aligned}\hat{h}_i[k] = f_i(&\mathbf{q}[k], q[k-1], \mathbf{q}[k-2], \\ &\mathbf{h}[k-1], \mathbf{h}[k-2], \mathbf{h}[k-3]), \qquad (13)\\ i = 1,2,3,\ & k = 4,\dots,N\end{aligned}$$

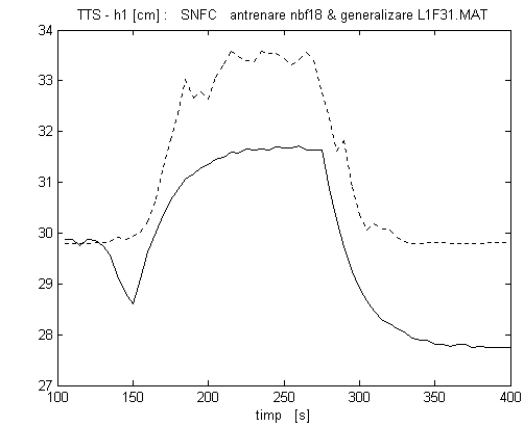
where $\mathbf{q}[k] = [q_i[k]]_{i=1,2}$ and $\mathbf{h}[k] = [h_i[k]]_{i=1,2,3}$ represent the vector of process inputs and the vector of process outputs, respectively. The CANFSs were used to determine the unknown functions f_i. They were trained using the algorithm described in Section 3. The resulted CANFS models consist of 3÷4 rules and the expansion orders of the FLNN local models are in the set $\{1, 2\}$.

Figures 4 and 5 illustrate some of the obtained experimental results. In these figures, the outputs of the process (solid line) are compared with the outputs of the corresponding identified CANFS models (dotted line).

Figures 4(a) and 5(a) illustrate the generalisation capability of the obtained CANFS models. In this case, process data corresponding to the normal behaviour, different than the training data, were used to feed the CANFS models. One observes that the developed CANFS models have good generalisation properties, i.e. are able to approximate with good precision, data different than the data used in the training stage corresponding to the normal behaviour of the process.



(a)



(b)

Fig.4 Three-tank system, output h_1 of the process and of the CANFS model trained for the [NB] class

Figures 4(b) and 5(b) present the response of the CANFS models developed for the normal behaviour of the process, when data corresponding to a faulty behaviour are presented to their input. One observes the big difference between the output of the process and the output of the CANFS models. This can be used as symptom of that fault, indicating that the current behaviour of the process is not normal.

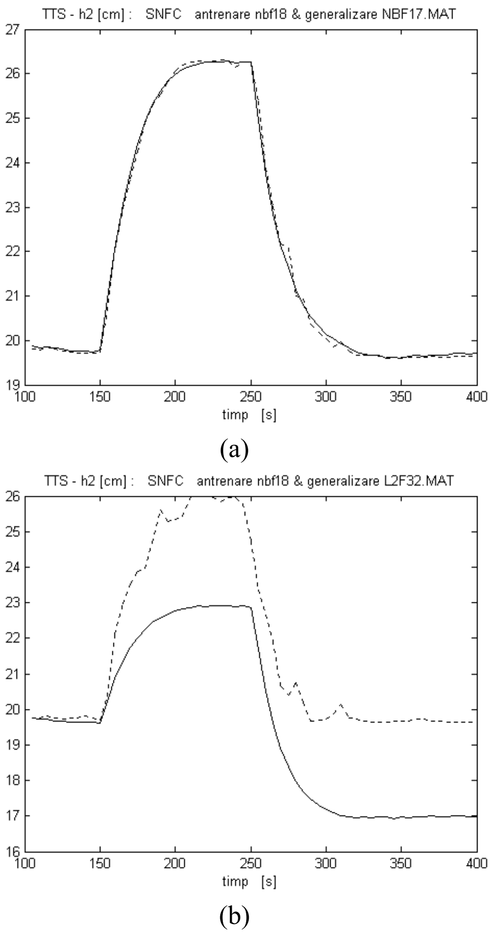


Fig.5 Three-tank system, output h_2 of the process and of the CANFS model trained for the [NB] class

The residuals in the form (12) are generated and a neural classifier is designed to analyse them. For this purpose, a MLP network with two layers of sigmoid neurons is used. The hidden layer of the neural classifier has 15 neurons. The obtained classification results are presented in Table 1.

Table 1: Three-tank system, performance of diagnosing systems (recognition rates [%])

Training	global	99.51
	normal	100
	faulty	99.34
Testing	global	96.7
	normal	97.35
	faulty	96.7

6. CONCLUSIONS

This paper investigates the development of a new neuro-fuzzy system with neural consequent part of the fuzzy rules and its application to component fault diagnosis (fault detection and fault isolation) of a Three-Tank System. The experimental results obtained by using the suggested neuro-fuzzy system reveal its good performances of approximation and generalisation. This application of fault diagnosis leads to good results, as reflected in a recognition rate of around 97%.

Further research will investigate the development of a new class of co-active neuro-fuzzy systems with internal dynamic elements and their application to fault detection and isolation. This will allow for an increase of the generalisation performances of the CANFS.

REFERENCES

Chen,C.L.P. and J.Z.Wan, A Rapid Learning and Dynamic Stepwise Updating Algorithm for Flat Neural Networks and the Application to Time-Series Prediction, *IEEE Trans. on Systems, Man, and Cybernetics – part B: Cybernetics*, **29**, 1, pp.62-72, (1999).

Chiu, S.L. (1994), A Cluster Estimation Method with Extension to Fuzzy Model Identification, *Proc. of the 3rd IEEE Int.Conf. on Fuzzy Systems*, Orlando, Florida, USA, 1240-1245.

Jang, J.-S. R. (1995), Neuro-Fuzzy Modelling and Control, *Proceedings of IEEE*, **83**, *3*, 378 – 405.

Marcu, T. (1996), Pattern recognition techniques using fuzzily labeled data for process fault diagnosis, *Int. Journ. of Appl. Math. & Comp. Science*, **6**, (4), 815 – 840.

Marcu,T., L.Mirea and P.M.Frank, Development of Dynamic Neural Networks with Application to Observer-based Fault Detection and Isolation, *International Journal of Applied Mathematics and Computer Science*, **9**, 3, pp.547-570, (1999).

Marcu, T., L. Mirea, P.M. Frank & H.D. Kochs (2001), System identification and fault diagnosis using dynamic functional-link neural networks, *European Control Conference ECC'2001*, Porto, Portugal, CD-ROM Proceedings.

Mirea, L. & T. Marcu (2002), System identification using a neuro-fuzzy technique, *2nd European Conference on Intelligent Systems and Technologies ECIT'2002*, Iaşi, Romania, CD-ROM Proceedings.

Pao,Y.H., G.H. Park and D.J.Sobajic, Learning and Generalisation Characteristics of the Random Vector Functional-Link Net, *Neurocomputing*, **6**, pp.163-180, (1994).

Patra, J.C. and R.N. Pal, A Functional-Link Artificial Network for Adaptive Channel Equalisation, *Signal Processing*, **43**, pp.181-195, (1995).

Patra, J.C., R.N. Pal, B.N. Chatterji and G. Panda, Identification of Non-linear Dynamic Systems Using Functional-Link Artificial Neural Networks, *IEEE Trans. on Systems, Man, and Cybernetics – part B: Cybernetics*, **29**, 2, pp.254-262, (1999).

Patton, R.J., R.Clark & P.M.Frank (2000), *Issues of Fault Diagnosis for Dynamic Systems*, Springer-Verlag New York, Inc.

Takagi, S. & M. Sugeno (1985), Fuzzy identification of systems and its application to modelling and control, *IEEE Trans. Systems, Man & Cybernetics*, **15**, pp. 116-132.

Zhang, J. & J. Morris (1996), Process modelling and fault diagnosis using fuzzy neural networks, *Fuzzy Sets and Systems*, **79**, 127-140.