Int. J. Appl. Math. and Comp. Sci., 1999, Vol.9, No.3, 547–570

DEVELOPMENT OF DYNAMIC NEURAL NETWORKS WITH APPLICATION TO OBSERVER-BASED FAULT DETECTION AND ISOLATION

TEODOR MARCU*, LETITIA MIREA*
PAUL M. FRANK**

The paper suggests a neural-network approach to the design of robust fault diagnosis systems. The main emphasis is placed upon the development of neural observer schemes. They are built based on dynamic neural networks, i.e. dynamic multi-layer perceptrons with mixed structure. The goal is to achieve an adequate approximation of process outputs for known classes of the process behaviour. The obtained symptoms are then classified by means of static artificial nets. Appropriate decision mechanisms are designed for each type of observer schemes. An application to a laboratory process is included. It refers to component and instrument fault detection and isolation in a three-tank system.

Keywords: fault diagnosis, dynamic neural networks, system identification, static neural classifiers, three-tank system.

1. Introduction

The methods of fault diagnosis must enable a condition-based inspection and repair, in response to the call for fault-tolerance in automatic control systems (Frank and Köppen-Seliger, 1997). This is due to the increasing demand for safe and reliable operation of uncertain and complex dynamic systems. Robust methods of diagnosis are therefore required, in the face of existing measurement uncertainty, disturbances and incomplete knowledge (Patton, 1994). This implies the maximisation of the detectability and isolability of faults under the constraint of minimisation of the false alarm rate.

A fault diagnosis system has to perform two tasks, namely fault detection and fault isolation. The purpose of the former is to determine that a fault has occurred in the process. The latter has the purpose of locating the fault. In order to accomplish these tasks, information that reflects a change from the normal behaviour of the process has to be obtained. This is generally called symptom generation. Secondly,

* 'Gheorghe Asachi' Technical University of Iasi, Department of Automatic Control and Industrial Informatics, Blvd. D. Mangeron No. 53A, RO–6600 Iasi, Romania, e-mail: tmarcu@ac.tuiasi.ro.

** 'Gerhard Mercator' University of Duisburg, Department of Measurement and Control, Bismarckstr. 81 (BB), D-47048 Duisburg, Germany.

a logical decision-making on the time of occurrence and the location of the fault has to be made. This is generally called symptom evaluation or fault classification.

The techniques applied most frequently to process fault diagnosis are based on estimation, i.e. parameter identification and observer-based methods (Isermann and Ballé, 1997). These methods generate analytical symptoms that characterise the state of the monitored system at a given instant of time. Symptom generation based on output estimation means to use state or output observers. Different schemes of estimators have led to successful robust fault diagnosis (Frank, 1994; Wünnenberg, 1990). However, most approaches are only applicable to linear systems. Robust nonlinear observers have been developed for particular classes of nonlinear dynamic systems (Frank *et al.*, 1999; Patton, 1994).

Artificial Neural Networks (ANN's) have been suggested as a possible technique to cope with the robustness problem in fault detection and isolation (FDI). A number of ANN structures and learning strategies were studied (Ayoubi, 1996; Frank and Köppen-Seliger, 1997; Isermann and Ballé, 1997; Isermann *et al.*, 1997; Köppen-Seliger, 1997; Sorsa *et al.*, 1993). ANN's have been used as both predictors of dynamic nonlinear models and pattern classifiers. These approaches do not require an accurate model of the process, but need representative training data. Many problems, especially that of coping with dynamics are not yet satisfactorily solved and need further research (Frank and Köppen-Seliger, 1997).

The neural models used most often are the feed-forward perceptron building multi-layer networks, i.e. the multi-layer perceptron (MLP), and the radial basis function (RBF) (Cichocki and Unbehauen, 1993; Haykin, 1994; Isermann *et al.*, 1997). Both the networks are capable of approximating any nonlinear unique static function to an arbitrary desired accuracy. However, it should be noticed that the MLP with back-propagation (BP) learning provides a global method for the design of an ANN, whereas RBF learning provides a local method (Haykin, 1994). In this respect, RBF could produce better approximations of training data. Conversely, when untrained data are processed by the identified ANN models, better results are obtained with MLP/BP. This is due to the fact that the locality of RBF networks means that they possess good interpolation and bad extrapolation abilities (Liang and ElMaraghy, 1993).

These types of mapping are well-suited for pattern recognition applications, where both the input vector and the output one represent spatial patterns that are independent of time (Haykin, 1994). The introduction of explicit dynamics into these ANN's requires a spatial representation of time. Time delay units are used to learn a system's dynamics. Thus, the network is fed with current and delayed values of the process inputs and outputs. The number of time delay units requires that the order of the system's dynamics must be given beforehand. It must be equal to or greater than the plant order. Another approach is to provide the mapping network with dynamic properties that make it responsive to time varying signals, i.e. locally recurrent globally feed-forward networks (Tsoi and Back, 1994) or neural networks with internal dynamics (Isermann *et al.*, 1997). A comparison regarding the application of static and dynamic neural nets to the design of FDI systems is given in a previous paper (Marcu and Mirea, 1997).

In this respect, the paper investigates firstly dynamic MLP's used to approximate nonlinear dynamic models of a plant. Three types of generalised dynamic ANN's (Marcu *et al.*, 1997) are properly integrated here in order to obtain the best approximation of process outputs for known classes of the system behaviour. Neural schemes are then suggested as alternative approaches to the well-known observer-based schemes (Frank, 1994; Wünnenberg, 1990). Neural observers are developed for component and instrument FDI. In this way, one tries to ensure active robustness when symptoms are generated. Further on, the passive robustness of the diagnosis subsystem is ensured, in the stage of decision-making, by means of static ANN's. They are used as pattern classifiers that evaluate those symptoms. An application to a laboratory process is included. Component fault diagnosis (CFD) and instrument fault diagnosis (IFD) of a three-tank system (ȧmira, 1993) are presented in a comparative study. This demonstrates the effectiveness of the suggested approaches.

2. Dynamic Neural Networks

The approach based on static ANN's leads to quasi-dynamic models (Ayoubi, 1996). The used neural net remains a static approximator, all of whose free parameters have fixed values (Haykin, 1994). The dimension of the input space of the network increases, depending on the number of available process data and the number of used past values. Instead, the main characteristic of dynamic ANN's is that they have memory. While limited forms of time-varying behaviours can be handled by using feed-forward networks and tapped delay lines, dynamic networks offer a much richer set of possibilities for representing the necessary internal states (Williams and Zipser, 1990). Since their internal state representation is adaptive rather than fixed, they are capable of preserving the state over certain periods of time. Thus, the network processes multi-inputs and does not require past values of the process measurements.

For fault diagnosis, the goal of 'best approximation' means to capture the dynamic behaviour of the process by learning the general trend of target values and filtering the noise. On the other hand, the artificial nets must have a minimal structure, in order to allow for a fast evaluation in the active stage of symptom generation. The interconnected or recurrent networks are able to learn time series, but they cannot represent input-output relations as easily as multi-layer networks do (Yokohama *et al.*, 1992). Therefore, three typical structures of dynamic MLP's (DMLP's) are considered and briefly described in the sequel. They have been generalised and comparatively applied to symptom generation for CFD in (Marcu *et al.*, 1997).

A DMLP with synaptic generalised filters (DMLP_SGF) has each synapse represented by an auto-regressive moving-average (ARMA) filter with different orders for denominator, n_A, and numerator, m_B. It generalises the architecture for which a finite-duration impulse response filter is used (Haykin, 1994). This generalisation leads to a DMLP with infinite-duration impulse response (IIR) filters used to replace the synaptic weights (Back and Tsoi, 1991; 1992), i.e. DMLP with local synapse feedback (Tsoi and Back, 1994). The DMLP with internal generalised filters (DMLP_IGF) introduces some dynamics into the transfer functions by integrating an ARMA filter within the neurons, i.e. before the activation function. It generalises the

DMLP net (Ayoubi, 1996), where the weighted sum (activation input) corresponding to each neuron acts like an IIR filter with equal orders for denominator, n_D, and numerator, m_E. This is the DMLP with local activation feedback (Isermann *et al.*, 1997). Finally, the DMLP with connectionist hidden layer (DMLP_CHL) has a partially recurrent structure, where only the hidden units are interconnected. It generalises the net presented in (Yokohama *et al.*, 1992), by accommodating bias terms for each activation function. This type of DMLP may be interpreted as a network with partially local output feedback (Tsoi and Back, 1994; Isermann *et al.*, 1997).

2.1. Dynamic MLP with Mixed Structure

The structures described previously are properly integrated into an ANN with mixed architecture, as illustrated by Fig. 1. The implementation of this structure is achieved in such a manner that one can select either a basic architecture or a combination of them. The DMLP *with mixed structure* (DMLP_MIX) and two layers is considered in the sequel. It is designed to approximate a multi-input single-output (MISO) dynamic process. The ANN has P inputs $u_p[k]$, $p = 1, \dots, P$, and one output $y^o[k]$, where $[k]$ denotes the sampling time instant k. The fully integrated structure is described in the following. The superscript/subscript h stands for the first (hidden) layer, and the superscript/subscript o denotes the second (output) layer. The next binary variables indicate the presence (value of 1) or absence (value of 0) of a dynamic structure: i_{SGF} for synaptic filters, i_{IGF} for internal filters, and i_{CHL} for the connectionist hidden layer.

The hidden layer has S neurons. The generic neuron s, $s = 1, \dots, S$ has the following description:

- inputs: $u_p[k]$, $p = 1, \dots, P$;
- synaptic weights:

$$w^h_{s,p}[k] := b^h_{s,p,0} u_p[k] + i_{\mathrm{SGF}} \left\{ \sum_{j=1}^{m_B} b^h_{s,p,j} u_p[k-j] - \sum_{i=1}^{n_A} a^h_{s,p,i} w^h_{s,p}[k-i] \right\} \tag{1}$$

- weighted input:

$$x^h_s[k] := \sum_{p=1}^{P} w^h_{s,p}[k] \tag{2}$$

- connectionist input:

$$\tilde{x}^h_s[k] := x^h_s[k] + i_{\mathrm{CHL}} \left\{ \sum_{j=1}^{S} c^h_{s,j} y^h_j[k-1] \right\} \tag{3}$$

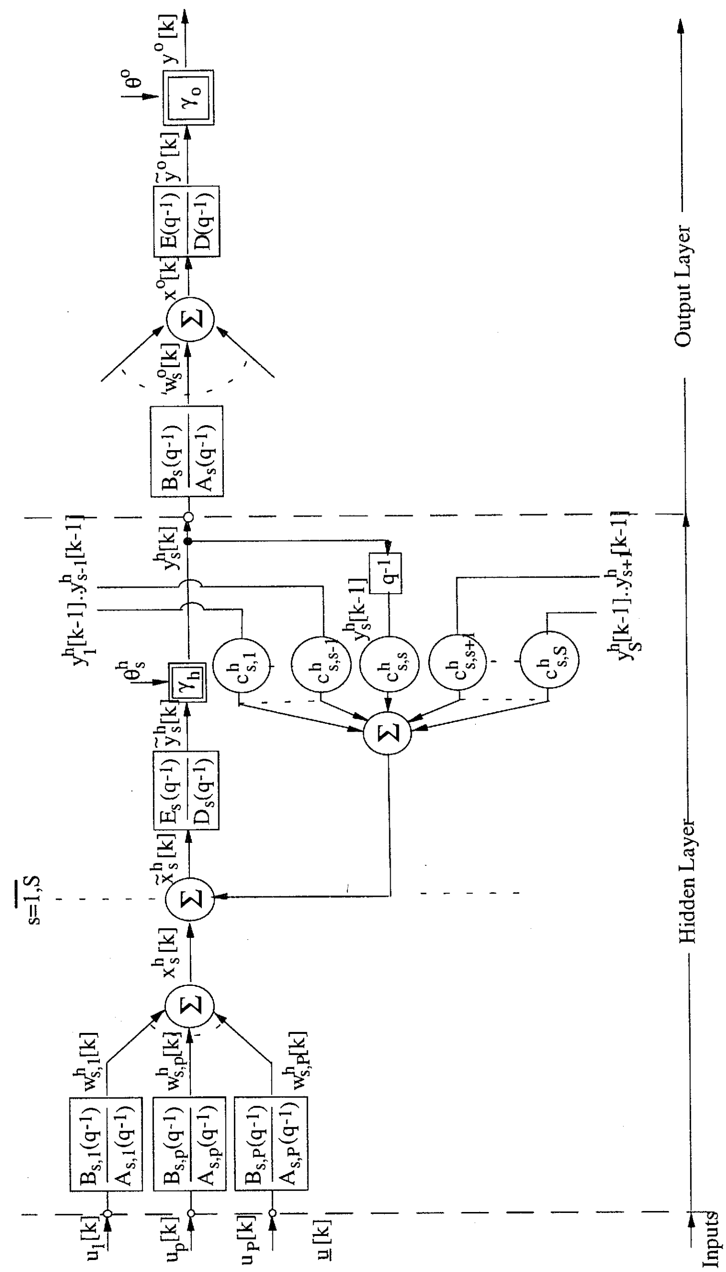


Fig. 1. The dynamic multi-layer perceptron with mixed structure: P inputs, one hidden layer with S neurons and one output; q^{-1} stands for the linear operator of time shifting, A., B., D. and E. are polynomials.

- output of internal filter:

$$\tilde{y}_s^h[k] := e_{s,0}^h \tilde{x}_s^h[k] + i_{\mathrm{IGF}} \left\{ \sum_{j=1}^{m_E} e_{s,j}^h \tilde{x}_s^h[k-j] - \sum_{i=1}^{n_D} d_{s,i}^h \tilde{y}_s^h[k-i] \right\} \tag{4}$$

- input of activation function with bias term θ_s^h:

$$z_s^h[k] := \tilde{y}_s^h[k] + \theta_s^h \tag{5}$$

- output (of activation function γ_h):

$$y_s^h[k] := \gamma_h\left(z_s^h[k]\right) \tag{6}$$

The hyperbolic tangent is usually considered as the activation function:

$$\gamma_h(z) := \frac{(e^z - e^{-z})}{(e^z + e^{-z})} \tag{7}$$

The derivative of this function is further used in the learning process:

$$\gamma_h(z) := \frac{4}{(e^z + e^{-z})^2} = 1 - \gamma_h^2(z) \tag{8}$$

The output layer has one neuron and the following description:

- inputs: $y_s^h[k],\ s = 1, \ldots, S$
- synaptic weights:

$$w_s^o[k] := b_{s,0}^o y_s^h[k] + i_{\mathrm{SGF}} \left\{ \sum_{j=1}^{m_B} b_{s,j}^o y_s^h[k-j] - \sum_{i=1}^{n_A} a_{s,i}^o w_s^o[k-i] \right\} \tag{9}$$

- weighted input:

$$x^o[k] := \sum_{s=1}^{S} w_s^o[k] \tag{10}$$

- output of internal filter:

$$\tilde{y}^o[k] := e_0^o x^o[k] + i_{\mathrm{IGF}} \left\{ \sum_{j=1}^{m_E} e_j^o x^o[k-j] - \sum_{i=1}^{n_D} d_i^o \tilde{y}^o[k-i] \right\} \tag{11}$$

- input of activation function with bias term θ^o:

$$z^o[k] := \tilde{y}^o[k] + \theta^o \tag{12}$$

- output (of activation function γ_o):

$$y^o[k] := \gamma_o(z^o[k]) \tag{13}$$

The activation function given in (7) can be used in the output layer as well. Another choice is to consider the linear activation function:

$$\gamma_o(z) := z \tag{14}$$

If the synaptic filters are not included, $i_{\mathrm{SGF}} = 0$, then eqns. (1) and (9) lead to constant synaptic weights, respectively:

$$\begin{aligned} w^h_{s,p}[k] &\equiv w^h_{s,p} := b^h_{s,p,0} \\ w^o_s[k] &\equiv w^o_s := b^o_{s,0} \end{aligned} \tag{15}$$

If the internal filters are not included, $i_{\mathrm{IGF}} = 0$, then the following relationships are used, instead of eqns. (4) and (11), respectively:

$$\begin{aligned} e^h_{s,0} &:= 1, \qquad \tilde{y}^h_s[k] \equiv \tilde{x}^h_s[k] \\ e^o_0 &:= 1, \qquad \tilde{y}^o[k] \equiv x^o[k] \end{aligned} \tag{16}$$

If the connectionist hidden layer is not considered, $i_{\mathrm{CHL}} = 0$, then the connectionist input given in eqn. (3) coincides with the weighted input from eqn. (2):

$$\tilde{x}^h_s[k] \equiv x^h_s[k] \tag{17}$$

Finally, if none of the dynamic structures is considered, $i_{\mathrm{SGF}} = i_{\mathrm{IGF}} = i_{\mathrm{CHL}} = 0$, then a static MLP is obtained.

2.2. Dynamic BP Learning

For a given structure with S hidden neurons, the parameters of the network are the filters' coefficients and bias terms. They are determined with an extended, dynamic back-propagation (BP) algorithm. It represents an adaptation of the exact gradient following algorithm of temporal supervised learning (Williams and Zipser, 1990). To describe its basic principle, the following notation is used: $y_d[k]$ represents the desired output of the net, η denotes the parameter of learning rate, and χ stands for the (parameter) states within the dynamic neurons.

Given N input-output data pairs:

$$\{\boldsymbol{u}[k], y_d[k]\}, \quad \boldsymbol{u}[k] := [u_p[k]]_{p=1,\dots,P}, \quad k = 1,\dots,N \tag{18}$$

each parameter, generally denoted by ξ, is adapted every learning epoch by the relationships:

$$\xi_{\text{new}} := \xi_{\text{old}} + \Delta\xi, \quad \Delta\xi := \sum_{k=1}^{N} \Delta\xi[k], \quad \Delta\xi[k] := -\eta\frac{\partial E[k]}{\partial \xi} \tag{19}$$

where $E[k]$ represents the square error function at time k:

$$E[k] := \frac{1}{2}\left(y_d[k] - y^o[k]\right)^2 \tag{20}$$

A learning epoch represents an entire pass through all of the input training vectors given in the relationship (18). The parameters are changed at the end of that epoch, that is the batch learning mode (Demuth and Beale, 1996). While BP uses the backward propagation to compute the error gradient, the introduction of parameters' states allows for a forward propagation of the activity gradient function. This represents the approach of real-time recurrent learning (Williams and Zipser, 1990).

The adaptation of parameters starts from the output layer. The following relationships are used:

$$\begin{aligned} \delta^o[k] &:= -\frac{\partial E[k]}{\partial z^o} = \left(y_d[k] - y^o[k]\right)\gamma_o'\left(z^o[k]\right) \\ \Delta\theta^o[k] &= \eta\delta^o[k] \end{aligned} \tag{21}$$

If $i_{\text{IGF}} = 1$, then

$$\begin{aligned} \chi_{d_i^o}[k] &:= \frac{\partial \tilde{y}^o[k]}{\partial d_i^o} = -\tilde{y}^o[k-i] - \sum_{l=1}^{n_D} d_l^o \chi_{d_i^o}[k-l], \quad i = 1,\dots,n_D \\ \Delta d_i^o[k] &= \eta\delta^o[k]\chi_{d_i^o}[k] \end{aligned} \tag{22}$$

$$\begin{aligned} \chi_{e_j^o}[k] &:= \frac{\partial \tilde{y}^o[k]}{\partial e_j^o} = x^o[k-j] - \sum_{i=1}^{n_D} d_i^o \chi_{e_j^o}[k-i], \quad j = 0,1,\dots,m_E \\ \Delta e_j^o[k] &= \eta\delta^o[k]\chi_{e_j^o}[k] \end{aligned} \tag{23}$$

Further on, one computes

$$\chi_{\tilde{y}^o}[k] := \frac{\partial \tilde{y}^o[k]}{\partial x^o} = e_0^o + i_{\text{IGF}}\left\{\sum_{j=1}^{m_E} e_j^o - \sum_{i=1}^{n_D} d_i^o \chi_{\tilde{y}^o}[k-i]\right\}$$

If $i_{\text{SGF}} = 1$, then

$$\begin{aligned} \chi_{a_{s,i}^o}[k] &:= \frac{\partial w_s^o[k]}{\partial a_{s,i}^o} = -w_s^o[k-i] - \sum_{l=1}^{n_A} a_{s,l}^o \chi_{a_{s,i}^o}[k-l], \quad i = 1,\dots,n_A \\ \Delta a_{s,i}^o[k] &= \eta\delta^o[k]\chi_{\tilde{y}^o}[k]\chi_{a_{s,i}^o}[k] \end{aligned} \tag{24}$$

$$\begin{aligned} \chi_{b_{s,j}^o}[k] &:= \frac{\partial w_s^o[k]}{\partial b_{s,j}^o} = y_s^h[k-j] - \sum_{i=1}^{n_A} a_{s,i}^o \chi_{b_{s,j}^o}[k-i], \quad j = 0,1,\dots,m_B \\ \Delta b_{s,j}^o[k] &= \eta\delta^o[k]\chi_{\tilde{y}^o}[k]\chi_{b_{s,j}^o}[k] \end{aligned} \tag{25}$$

else eqn. (25) reduces to

$$\Delta w_s^o[k] = \Delta b_{s,0}^o[k] = \eta \delta^o[k] \chi_{\tilde{y}^o}[k] y_s^h[k]$$

The parameters involved in the hidden layer are adapted according to the following formulae

$$\chi_{y_s^h}[k] := \frac{\partial w_s^o[k]}{\partial y_s^h} = b_{s,0}^o + i_{\mathrm{SGF}} \left\{ \sum_{j=1}^{m_B} b_{s,j}^o - \sum_{i=1}^{n_A} a_{s,i}^o \chi_{y_s^h}[k-i] \right\}$$

$$\delta_s^h[k] := -\frac{\partial E[k]}{\partial z_s^h} = \delta^o[k] \chi_{\tilde{y}^o}[k] \chi_{y_s^h}[k] \gamma_h'\left(z_s^h[k]\right)$$

$$\Delta \theta_s^h[k] = \eta \delta_s^h[k] \tag{26}$$

If $i_{\mathrm{IGF}} = 1$, then

$$\chi_{d_{s,i}^h}[k] := \frac{\partial \tilde{y}_s^h[k]}{\partial d_{s,i}^h} = -\tilde{y}_s^h[k-i] - \sum_{l=1}^{n_D} d_{s,l}^h \chi_{d_{s,i}^h}[k-l], \quad i = 1, \dots, n_D$$

$$\Delta d_{s,i}^h[k] = \eta \delta_s^h[k] \chi_{d_{s,i}^h}[k] \tag{27}$$

$$\chi_{e_{s,j}^h}[k] := \frac{\partial \tilde{y}_s^h[k]}{\partial e_{s,j}^h} = \tilde{x}_s^h[k-j] - \sum_{i=1}^{n_D} d_{s,i}^h \chi_{e_{s,j}^h}[k-i], \quad j = 0, 1, \dots, m_E$$

$$\Delta e_{s,j}^h[k] = \eta \delta_s^h[k] \chi_{e_{s,j}^h}[k] \tag{28}$$

Further on, one computes

$$\chi_{\tilde{y}_s^h}[k] := \frac{\partial \tilde{y}_s^h[k]}{\partial \tilde{x}_s^h} = e_{s,0}^h + i_{\mathrm{IGF}} \left\{ \sum_{j=1}^{m_E} e_{s,j}^h - \sum_{i=1}^{n_D} d_{s,i}^h \chi_{\tilde{y}_s^h}[k-i] \right\}$$

If $i_{\mathrm{CHL}} = 1$, then

$$\chi_{q,c_{s,j}^h}[k] := \frac{\partial \tilde{x}_q^h[k]}{\partial c_{s,j}^h} = \sum_{i=1}^{S} c_{q,i}^h \gamma_h'\left(z_i^h[k-1]\right) \chi_{\tilde{y}_i^h}[k-1] \chi_{i,c_{s,j}^h}[k-1]$$

$$q = 1, \dots, S, \quad q \neq s$$

$$\chi_{s,c_{s,j}^h}[k] := \frac{\partial \tilde{x}_s^h[k]}{\partial c_{s,j}^h}$$

$$= y_j^h[k-1] + \sum_{q=1}^{S} c_{s,q}^h \gamma_h'\left(z_q^h[k-1]\right) \chi_{\tilde{y}_q^h}[k-1] \chi_{q,c_{s,j}^h}[k-1]$$

$$\Delta c_{s,j}^h[k] = \eta \delta_s^h[k] \chi_{\tilde{y}_s^h}[k] \chi_{s,c_{s,j}^h}[k] \tag{29}$$

If $i_{\mathrm{CHL}} = 1$, then

$$\chi_{q,a^h_{s,p,i}}[k] := \frac{\partial \tilde{x}^h_q[k]}{\partial a^h_{s,p,i}} = \sum_{l=1}^{S} c^h_{q,l}\gamma'_h\left(z^h_l[k-1]\right)\chi_{\tilde{y}^h_l}[k-1]\,\chi_{l,a^h_{s,p,i}}[k-1]$$

$$q = 1,\dots,S, \quad q \neq s$$

If $i_{\mathrm{SGF}} = 1$, one proceeds further by

$$\chi_{a^h_{s,p,i}}[k] := \frac{\partial w^h_{s,p}[k]}{\partial a^h_{s,p,i}} = -w^h_{s,p}[k-i] - \sum_{l=1}^{n_A} a^h_{s,p,l}\chi_{a^h_{s,p,i}}[k-l], \quad i = 1,\dots,n_A$$

$$\chi_{s,a^h_{s,p,i}}[k] := \frac{\partial \tilde{x}^h_s[k]}{\partial a^h_{s,p,i}} = \chi_{a^h_{s,p,i}}[k] + i_{\mathrm{CHL}}\left\{\sum_{q=1}^{S} c^h_{s,q}\gamma'_h\left(z^h_q[k-1]\right)\chi_{\tilde{y}^h_q}[k-1]\chi_{q,a^h_{s,p,i}}[k-1]\right\}$$

$$\Delta a^h_{s,p,i}[k] = \eta\delta^h_s[k]\,\chi_{\tilde{y}^h_s}[k]\,\chi_{s,a^h_{s,p,i}}[k] \tag{30}$$

If $i_{\mathrm{CHL}} = 1$, then

$$\chi_{q,b^h_{s,p,j}}[k] := \frac{\partial \tilde{x}^h_q[k]}{\partial b^h_{s,p,j}} = \sum_{l=1}^{S} c^h_{q,l}\gamma'_h\left(z^h_l[k-1]\right)\chi_{\tilde{y}^h_l}[k-1]\,\chi_{l,b^h_{s,p,j}}[k-1]$$

$$q = 1,\dots,S, \quad q \neq s$$

Further on, one proceeds by

$$\chi_{b^h_{s,p,j}}[k] := \frac{\partial w^h_{s,p}[k]}{\partial b^h_{s,p,j}} = u_p[k-j] - \sum_{i=1}^{n_A} a^h_{s,p,i}\chi_{b^h_{s,p,j}}[k-i], \quad j = 0,1,\dots,m_B$$

$$\chi_{s,b^h_{s,p,j}}[k] := \frac{\partial \tilde{x}^h_s[k]}{\partial b^h_{s,p,j}} = \chi_{b^h_{s,p,j}}[k] + i_{\mathrm{CHL}}\left\{\sum_{q=1}^{S} c^h_{s,q}\gamma'_h\left(z^h_q[k-1]\right)\chi_{\tilde{y}^h_q}[k-1]\,\chi_{q,b^h_{s,p,j}}[k-1]\right\}$$

$$\Delta b^h_{s,p,j}[k] = \eta\delta^h_s[k]\,\chi_{\tilde{y}^h_s}[k]\,\chi_{s,b^h_{s,p,j}}[k] \tag{31}$$

The learning algorithm starts with small random values for the net parameters, except for the filter coefficients of denominators. They are initialised to zeros to support a stable learning. The internal states χ are initialised to zeros. To implement this approach, new functions for the Neural Network Toolbox (Demuth and Beale, 1996) have been developed, for both determination of network parameters and for net evaluation. In addition, the mechanisms of variable parameter of learning rate and momentum term have been considered (Cichocki and Unbehauen, 1993).

The former mechanism attempts to keep the learning step size as large as possible, while maintaining a stable learning. When the learning rate is too high to guarantee a decrease in the error, it gets decreased until stable learning resumes. The latter mechanism makes parameter changes equal to the sum of a fraction of the last epoch change and the new one achieved by the rule given by eqn. (19). This allows the network to respond not only to the local gradient, but also to recent trends in the error surface.

3. Neural Design of the FDI System

For the generation of symptoms, neural networks replace the analytical model that describes the process (Frank and Köppen-Seliger, 1997). Instead of a multi-input multi-output structure, an ANN model for each system output is identified (Sorsa *et al.*, 1993). Since the closed loop of the control system tends to hide the faults, both inputs and outputs of the process are used as inputs of the ANN. Thus, several small ANN models are identified for a known class of process behaviour. These models are used for the separate estimation of process output signals in an observer-like arrangement. It should, however, be noticed that for FDI purposes, one needs representative models of the plant. This means that one needs only that part of the model that reflects the faults of interest and, with respect to robustness, it is not or only weakly affected by disturbances and modelling uncertainty. In this way, the models for FDI can be simpler than those for control (Frank *et al.*, 1997).

The next stage of decision-making can be seen as a classification problem. For fault diagnosis, this means to match each pattern of the symptom vector with one of the pre-assigned classes of faulty behaviour, if available, and the fault-free class, respectively (Frank and Köppen-Seliger, 1997). The decision is usually based on the prediction error between the process measurements and the predicted outputs, i.e. the residual signals. Fixed and/or adaptive thresholds are then used inside the detection logic. A more robust decision is achieved by using an ANN as the pattern classifier (Marcu and Mirea, 1997). It has as inputs the residual signals and must produce a pre-assigned outputs characteristic to each known class of process behaviour. The uncertainty in the 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.

3.1. Symptom Generation

The goal of observer schemes used in FDI is to generate structured sets of residuals that enable a unique fault diagnosis. The general Fault Detection Observer scheme consists of a number of observers. Each must be sensitive to different faults or set of faults that have to be detected and isolated (Wünnenberg, 1990). This may be accomplished by driving the observers by different sets of inputs and outputs of the process (Frank, 1994). Neural approaches to classical observer-based schemes are introduced in the sequel. One considers a process with I inputs $q_i[k]$, $i = 1, \ldots, I$, and O outputs $h_j[k]$, $j = 1, \ldots, O$, all known at sampling time k.

The *neural simplified observer scheme* (NSOS) consists of a number of MISO neural networks. Each of them is driven by all inputs and outputs of the process. Each net estimates one output $h_j[k]$ of the system:

$$\hat{h}_j\,[k] := f_{\mathrm{NSOS}_j}\big(\boldsymbol{q}\,[k]\,,\boldsymbol{h}\,[k-1]\big), \quad j = 1, \ldots, O \tag{32}$$

$$\boldsymbol{q}\,[k] := \big[q_i\,[k]\big]_{i=1,\ldots,I}, \quad \boldsymbol{h}\,[k-1] := \big[h_j\,[k-1]\big]_{j=1,\ldots,O}$$

The resulted bank of ANN's approximates all outputs of the process. The training of the nets is based on the system data corresponding to the normal behaviour. The following residuals are then generated:

$$\epsilon_j\,[k] := h_j\,[k] - \hat{h}_j\,[k]\,, \quad j = 1, \ldots, O \tag{33}$$

If appropriate data are available for different classes of faulty behaviour, the principle of NSOS is extended to them as well. In this way, a *neural multiple observer scheme* (NMOS) results. The following approximations are therefore performed:

$$\hat{h}_{j,c}\,[k] := f_{\mathrm{NMOS}_{j,c}}\big(\boldsymbol{q}_c[k], \boldsymbol{h}_c[k-1]\big) \tag{34}$$

where c denotes a particular class of behaviour, $c = 1, \ldots, C$, and j denotes a process output, $j = 1, \ldots, O$. The basic concept is that the c-th set out of the C groups of residuals is designated to be sensitive to all but the c-th fault, since the corresponding observer is designed to reproduce the c-th class of system behaviour. Here, C represents the number of classes of process behaviour that are taken into consideration. This scheme is especially designed for the diagnosis of the system's components (Marcu and Mirea, 1997; Marcu *et al.*, 1997). The obtained residuals are then given by:

$$\epsilon_{j,c}\,[k] := h_j\,[k] - \hat{h}_{j,c}\,[k]\,, \quad j = 1, \ldots, O, \quad c = 1, \ldots, C \tag{35}$$

Moreover, some of the ANN inputs may be removed, in order to generate residuals that allow for the isolation of a specific fault. In this way, neural approaches to the well-known dedicated observer scheme (DOS) and generalised observer scheme (GOS) (Frank, 1994; Wünnenberg, 1990) can be developed. The neural variants are described in the sequel.

For the *neural dedicated observer scheme* (NDOS), an observer is dedicated to each instrument or component of a plant. It is driven by the process inputs and the output of the unit to be supervised. The observer estimates as many output signals as possible. The neural implementation of each observer of DOS is based on a number of MISO nets. Each net estimates one output of the system:

$$\hat{h}_{j,l}\,[k] := f_{\mathrm{NDOS}_{j,l}}\big(\boldsymbol{q}\,[k]\,, h_l\,[k-1]\big), \quad j = 1, \ldots, O, \quad l = 1, \ldots, O \tag{36}$$

where the first index, j, refers to the approximated output, and the second index, l, stands for the driving output signal, i.e. it counts for the index of NDOS. This is done for all sensors/components. The training of the nets is based on the process data corresponding to the normal behaviour. The following residual signals are then computed:

$$\epsilon_{j,l}\left[k\right] := h_j\left[k\right] - \hat{h}_{j,l}\left[k\right], \quad j = 1,\ldots,O, \quad l = 1,\ldots,O \tag{37}$$

The *neural generalised observer scheme* (NGOS) consists of as many observers as process outputs are available. Each observer is driven by the process inputs and all outputs but the output of the unit to be supervised:

$$\boldsymbol{h}_{\mathrm{NGOS}_l}\left[k-1\right] := \left[h_j\left[k-1\right]\right]_{j=1,\ldots,O;\ j\neq l} \tag{38}$$

The observer estimates as many output signals as possible. The neural implementation of each observer of GOS is based on a number of MISO nets. Each net estimates one output of the system:

$$\hat{h}_{j,l}\left[k\right] := f_{\mathrm{NGOS}_{j,l}}\big(\boldsymbol{q}[k], \boldsymbol{h}_{\mathrm{NGOS}_l}[k-1]\big), \quad j = 1,\ldots,O, \quad l = 1,\ldots,O \tag{39}$$

where the first index, j, refers to the approximated output, and the second index, l, counts for the index of NDOS. The training of the nets is based on the process data corresponding to the normal behaviour. Finally, the residuals are obtained, as given by eqn. (37).

3.2. Symptom Evaluation

The successive layers of a static MLP with sigmoid neurons

$$\gamma(z) := \frac{1}{1+e^{-z}} \tag{40}$$

and BP rule of learning carry out a sequence of mappings. This happens until one finds a representation in a suitable space, where the desired separation is possible. Such an ANN is a non-parametric nonlinear classifier. It maps the patterns from the feature space into a decision space. The patterns belonging to a class are made to cluster there around a pre-selected point, optimally chosen (Marcu, 1996). In this way, the decision regions determined by the ANN classifier in the feature space are complex, including those that are linearly inseparable, non-convex and disconnected. Such a classifier is to be used to cope with the problem of passive robustness in fault diagnosis.

In the approach, one uses one static ANN as the pattern classifier. It has as inputs all signals obtained by subtracting the neural approximations of all observers from the corresponding process measurements, i.e. the residuals given by one of eqns. (33), (35) and (37). It is therefore referred to as the *fully connected classifier* (FCC). A common choice of the vectors of decision space is based on the set of vertices of Euclidean vectors. An optimised choice of the target vectors is suggested in (Marcu, 1996; Marcu and Mirea, 1997). This approach can be applied to all neural schemes that have been introduced. When FCC processes the residuals given by NMOS, one selects the best fitting group of prediction models for either normal operation or one of the learnt faulty situations.

Another approach is suggested for the evaluation of residuals obtained from NDOS and NGOS, designed for IFD for instance. A static ANN is associated to each neural observer. Such a network is a binary classifier. Its target output has a low value, e.g. 0.1, when the corresponding residuals are not influenced by a sensor fault. A high value, e.g. 0.9, must be obtained otherwise. A bank of binary classifiers results. It is referred to as the *partially connected classifier* (PCC). Its outputs constitute a vector characteristic to each class of known behaviour. The decision vector associated to NDOS must have all components equal to 0.1 but the j-th component, when a fault occurs in sensor j:

$$\boldsymbol{t}_j = \begin{bmatrix} \underset{1}{0.1} & \cdots & 0.1 & \underset{j}{0.9} & 0.1 & \cdots & \underset{O}{0.1} \end{bmatrix}$$

The decision vector of NGOS must have all components equal to 0.9 but the j-th component, when a fault occurs in sensor j:

$$\boldsymbol{t}_j = \begin{bmatrix} \underset{1}{0.9} & \cdots & 0.9 & \underset{j}{0.1} & 0.9 & \cdots & \underset{O}{0.9} \end{bmatrix}$$

Irrespective of the neural scheme used, the decision vector must have all components equal to 0.1 in the fault-free case.

A *fault* is *detected and isolated*, if an unknown input pattern is mapped closest to one of the target vectors. The latter corresponds to the associated learnt class that reflects a fault. A fault is only detected if the input pattern is mapped far from all learned classes, that is the concept of reject option (Marcu, 1996).

The last concept is based on the computed distances between an actual output of the net and the target vectors. If these distances exceed certain threshold values, the input pattern belongs to an unknown class. The threshold values characterise the separability between the training classes. They are experimentally chosen based on the training set of the classifier.

If a *fault* is *only detected*, that is a (new) faulty situation, the synthesis of the classifier must only be reconsidered for further fault diagnosis. The neural observer schemes remain unchanged, excepting the NMOS for which a new observer might be added. The latter corresponds to the new (faulty) behaviour of the process.

4. Application

The applicability of suggested methodologies was studied with respect to the design of two model-based subsystems for monitoring the faults in components and sensors of a three-tank system. Although the dynamic modelling of the system under investigation is relatively simple, the resulted nonlinear model is only a limited approximation. This is evidenced in (Wünnenberg, 1990). An advanced technique of diagnosis is applied there, namely the unknown input fault detection observer (UIFDO). For instance, small leaks in tanks cannot be detected. This type of faults shows exactly the same response in symptoms as an inaccuracy in the description of the incoming mass flows. The present study suggests alternative approaches to robust FDI of that process, with respect to the modelling errors.

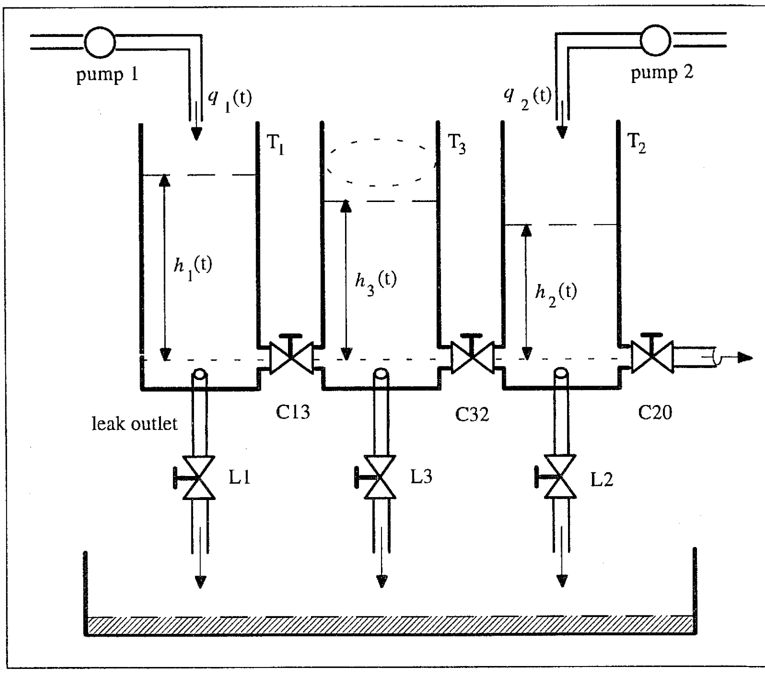


Fig. 2. The three-tank system.

4.1. Process Description

The experimental set-up 'Three-Tank System' (amira, 1993) consists of three cylindrical tanks with identical cross sections being filled with water, as shown in Fig. 2. The tanks are interconnected by circular pipes. All the three tanks are equipped with piezo-resistive pressure transducers for measuring the level of the liquid.

The system can be modelled conveniently by the mass balances of the tanks. The model is represented by three nonlinear differential equations of first order. It is used by an appropriate strategy implemented on a microcomputer. The water inlet is controlled by two pumps that are driven by an electronic power device. In this way, the volume flows of lateral tanks (the two process inputs $q_1(t)$ and $q_2(t)$) are controlled such that the level in the corresponding tanks (two out of three process outputs, $h_1(t)$ and $h_2(t)$) can be pre-assigned independently. Here, t stands for the time variable. The third output of the process, that is the level $h_3(t)$ in the middle tank, is always a signal that is uncontrollable. The control strategy worked at a sampling rate $T_C = 0.1\,\mathrm{s}$.

4.2. Component Fault Diagnosis

The connecting pipes and tanks are additionally equipped with manually adjustable valves and outlets to simulate clogs and leaks. Seven classes of process behaviour were taken into consideration. They are the following:

- ⟨NB⟩: normal behaviour (valves C_{13}, C_{32}, C_{20} are open, outlets L_1, L_3, L_2 are closed);
- ⟨L_1⟩: leakage in tank T_1 (outlet L_1 is partially open);
- ⟨L_3⟩: leakage in tank T_3 (outlet L_3 is partially open);
- ⟨L_2⟩: leakage in tank T_2 (outlet L_2 is partially open);
- ⟨C_{13}⟩: clogging in pipe between tanks T_1 and T_3 (valve C_{13} is partially closed);
- ⟨C_{32}⟩: clogging in pipe between tanks T_3 and T_2 (valve C_{32} is partially closed);
- ⟨C_{20}⟩: clogging in the outlet of tank T_2 (valve C_{20} is partially closed).

For the experiments, the reference values of the liquid levels were changed pulse-wise with different magnitude and duration for each controlled tank. A test period of 400 s was considered. Thirty-five experiments were performed for each class of behaviour in a period of a month, in order to take into consideration the influence of the plant environment. The input-output data of the process were sampled at every $T_S = 5\,\mathrm{s}$ during the test period.

Each of the ANN's, corresponding to system outputs, was trained with representative input-output data for each known class of the process behaviour. Thus, an NMOS was synthesised. One used the set of process data, among the 35 sets available, that had the corresponding values for the steady-states closest to the mean values of each class.

The following ANN models were identified for one-step ahead prediction:

$$\hat{h}_{j,c}[k] := f_{\mathrm{NMOS}_{j,c}}\left(\boldsymbol{q}_c[k_u], \boldsymbol{h}_c[k-1]\right), \quad j = 1,3,2, \quad c = 1,\dots,7$$

where $\hat{h}_{j,c}$ denotes the estimated value of a liquid level, $\boldsymbol{q}_c := [q_1\ q_2]$ denotes the vector of input flow rates, $\boldsymbol{h}_c := [h_1\ h_3\ h_2]$ denotes the vector of measured liquid levels, and c stands for one of the considered classes of process behaviour. The process outputs were known at sampling time $t = kT_S$, and the process inputs were known at sampling time $kT_S - T_C$. The last quantity is differentiated in the above formula by the normalised sampling time k_u. A systematic search for the best approximation achieved by the unknown nonlinear functions f was carried out. This involved all possible dynamic structures, i.e. the basic architectures of generalised dynamic ANN's (Marcu *et al.*, 1997) and their combinations. For each structure, one investigated ANN's with $S = 1,2,3$ hidden neurons and ARMA filters with orders of denominator and numerator in the set $\{0,1,2,3\}$. The nets had one hidden layer of hyperbolic tangent neurons, linear/hyperbolic tangent neurons in the output layer, and filters with variable orders, respectively. In general, the best results were obtained by using linear neurons in the output layer of DMLP's. Table 1 presents the structures of selected neural nets for NMOS.

Table 1. NMOS for CFD of a three-tank system.

Process Class	ANN Output	DMLP_MIX Structure					
		S	n_A	m_B	n_D	m_E	i_{CHL}
⟨NB⟩	$\hat{h}_1[k]$	1	2	0	3	0	1
	$\hat{h}_3[k]$	1			3	3	1
	$\hat{h}_2[k]$	2	0	2			1
⟨L_1⟩	$\hat{h}_1[k]$	2	0	2			1
	$\hat{h}_3[k]$	2	0	2			1
	$\hat{h}_2[k]$	1			3	0	1
⟨L_3⟩	$\hat{h}_1[k]$	2			1	2	1
	$\hat{h}_3[k]$	2			1	2	1
	$\hat{h}_2[k]$	2	2	0	0	2	0
⟨L_2⟩	$\hat{h}_1[k]$	2	2	0	0	2	0
	$\hat{h}_3[k]$	1	2	0			1
	$\hat{h}_2[k]$	1			1	2	1
⟨C_{13}⟩	$\hat{h}_1[k]$	1	2	0	0	3	1
	$\hat{h}_3[k]$	1			3	3	1
	$\hat{h}_2[k]$	1			3	0	1
⟨C_{32}⟩	$\hat{h}_1[k]$	2			1	2	1
	$\hat{h}_3[k]$	1	2	1			1
	$\hat{h}_2[k]$	2	0	2			1
⟨C_{20}⟩	$\hat{h}_1[k]$	1	2	0	3	0	1
	$\hat{h}_3[k]$	1			3	3	1
	$\hat{h}_2[k]$	1			3	0	1

Generally, the best results were obtained by using ANN's with mixed structure. A connectionist hidden layer was involved in almost all combinations. It seemed to improve the results of approximation due to its (partially) recurrent structure. For noisy data (e.g. the output h_1 of the process), the best results were obtained using the fully involved integrated structure. For smooth data (e.g. the output h_3 of the system), the combination of DMLP_CHL with either DMLP_SGF or DMLP_IGF seemed to produce the best approximation. In all these experiments, a reduced number of neurons in the hidden layer was sufficient, i.e. $S = 1, 2$. This allowed for a fast evaluation of the ANN outputs in the active stage of approximation. Figures 3 and 4 illustrate some of the results obtained.

For residual evaluation, one static MLP/BP net with two layers of sigmoid neurons was used. That FCC had 21 inputs, 21 hidden neurons, and an optimised number of 4 output neurons (Marcu, 1996). In the stage of fault classification, a global recognition rate around 90% was obtained. This characterised the recognised

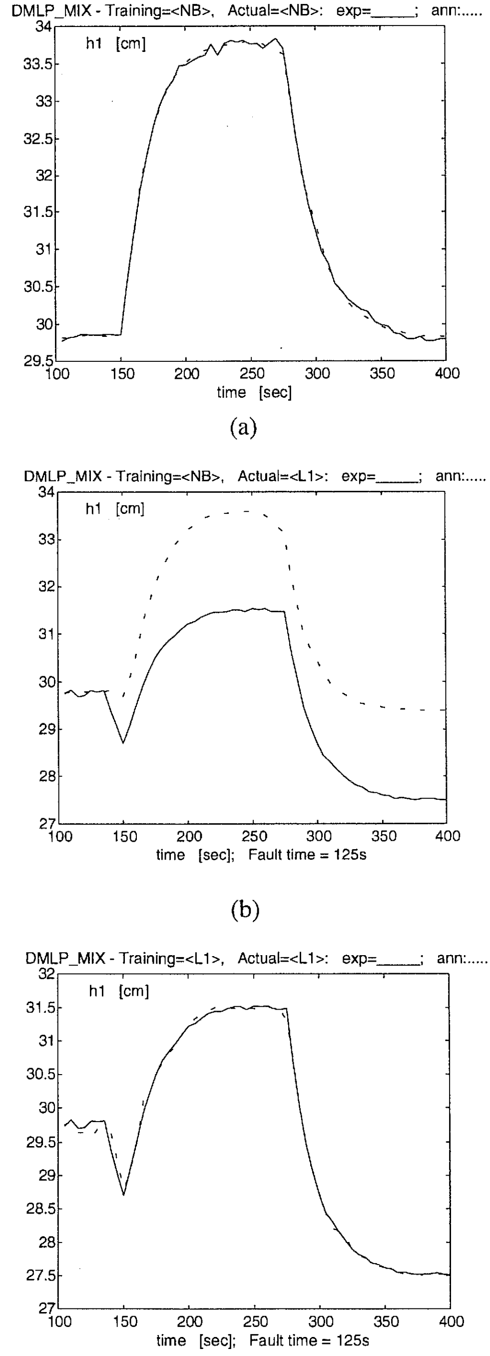
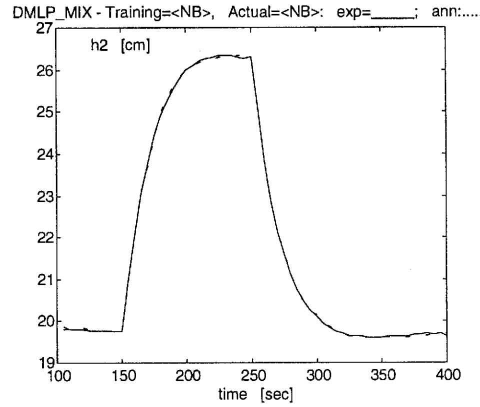
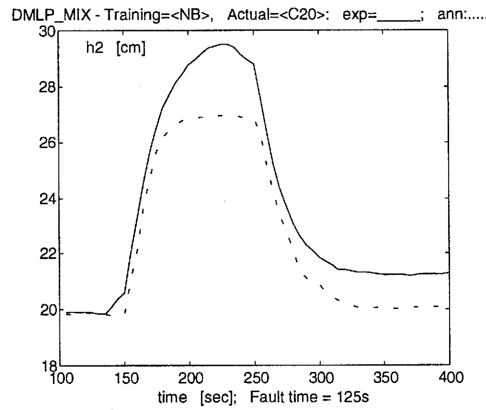


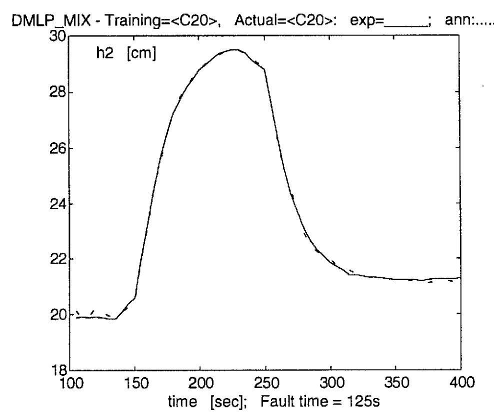
Fig. 3. Output h_1 of the process (solid line) and of the identified model (dotted line); training classes: $\langle$NB$\rangle$ (a), (b), $\langle L_1\rangle$ (c); actual classes: $\langle$NB$\rangle$ (a), $\langle L_1\rangle$ (b), (c).



(a)



(b)



(c)

Fig. 4. Output h_2 of the process (solid line) and of the identified model (dotted line); training classes: $\langle NB \rangle$ (a), (b), $\langle C_{20} \rangle$ (c); actual classes: $\langle NB \rangle$ (a), $\langle C_{20} \rangle$ (b), (c).

faults corresponding to the 35 sets of input-output data available for each class. This result indicates an accurate modelling of the fault mode dynamics. The detected faults corresponded to mass flows of about 10–20 ml/s. The approach based on a bank of UIFDOs detected only faults corresponding to mass flows of about 40 ml/s (Wünnenberg, 1990). The mechanism of reject option detected simulated faults in sensors as belonging to an unknown class. Finally, the attempt to use other neural observer schemes for symptom generation, e.g. either NDOS or NGOS, did not lead to better results.

4.3. Instrument Fault Diagnosis

Four classes of process behaviour were taken into consideration. They are the normal behaviour and incipient faults in each of the measuring instruments. The faults were artificially simulated by 10% reduction of each measured value of the liquid levels. For the experiments, the reference values of the liquid levels were changed step-wise with a different magnitude for each controlled tank, respectively. A test period of 300 s was considered. The input-output data of the process were sampled at every $T_S = 5$ s during the test period.

Each of the ANN's, corresponding to system outputs was trained by using the process data corresponding to the normal behaviour. The following neural observers were synthesised: NSOS, NDOS, and NGOS. The same systematic search for the best approximation was carried out as presented in the previous sub-section of the paper. The DMLP_MIX produced again the best approximations. The hidden layer contained hyperbolic tangent neurons and the output layer contained a linear neuron. The best results were obtained using a connectionist hidden layer in almost all combinations. In all these experiments, a reduced number of neurons in the hidden layer was sufficient as well. The structures of selected neural networks for NGOS are presented in Table 2.

Table 2. NGOS for IFD of the three-tank system.

ANN Inputs	ANN Output	DMLP_MIX Structure					
		S	n_A	m_B	n_D	m_E	i_{CHL}
NGOS$_1$:	$\hat{h}_1[k]$	1	0	2			1
$q_1[k_u]$, $q_2[k_u]$,	$\hat{h}_3[k]$	1	0	2	0	2	1
$h_2[k-1]$, $h_3[k-1]$	$\hat{h}_2[k]$	1	0	2			1
NGOS$_2$:	$\hat{h}_1[k]$	1	0	2			1
$q_1[k_u]$, $q_2[k_u]$,	$\hat{h}_3[k]$	1	0	2	0	2	1
$h_1[k-1]$, $h_2[k-1]$	$\hat{h}_2[k]$	1	0	2			1
NGOS$_3$:	$\hat{h}_1[k]$	3	0	2			1
$q_1[k_u]$, $q_2[k_u]$,	$\hat{h}_3[k]$	1	0	2			1
$h_1[k-1]$, $h_3[k-1]$	$\hat{h}_2[k]$	4			3	3	1

Static MLP/BP nets with two layers of sigmoid neurons were used for symptom evaluation. They were trained to discriminate among the above-mentioned four classes. The residuals corresponding to faulty behaviours were obtained by simulating the abnormal situations just before the change in process references. FCC's were synthesised to classify residuals obtained from all considered neural observer schemes. PCC's were synthesised to process residuals obtained from NDOS and NGOS.

Table 3. Overall recognition rates of neural observers for IFD.

Observer/	Training	Generalisation	
Classifier	(dynamic)	(static)	(dynamic)
NSOS / FCC	98.75 %	81.87 %	98.33 %
NDOS / FCC	100 %	62.63 %	94.16 %
NDOS / PCC	100 %	78.75 %	95 %
NGOS / FCC	100 %	96.88 %	95.83 %
NGOS / PCC	99.38 %	73.75 %	98.33 %

The robustness of the diagnosing subsystem was tested as well. One used process data corresponding to faults that were simulated in both the steady state of the process and the dynamic regime. In the latter case, the abnormal situations were considered after the change in the system's input references. Table 3 shows an overview of the performances obtained. Each recognition rate represents the mean value of those corresponding to the considered four classes of process behaviour. These experimental results have led to the choice of NGOS with FCC, due to its overall best performances.

The approach based on a bank of UIFDO's detected only accentuated faults in sensors (Wünnenberg, 1990). The decisions obtained were characterised by missing and false alarms. Finally, the mechanism of reject option detected the considered faults in components as belonging to an unknown class. Therefore, both the diagnosing systems produced correct fault alarms.

5. Concluding Remarks

The present paper suggests neural approaches to observer-based schemes in order to perform a robust diagnosis of process faults. The symptoms are generated by using dynamic neural networks with mixed structure. The residuals are then classified by means of static artificial nets. An application to a laboratory process is included. It offers, however, a properly limited environment before the complicated industrial plants are studied.

The study demonstrates that neural networks provide efficient tools for system modelling, identification and pattern recognition. The main advantage of the neural approach is that no mathematical model of the plant is needed. The robustness of the diagnosing systems is complementary ensured in both steps of symptom generation and evaluation. This has led to better performances of two diagnosing subsystems designed to detect incipient faults in components and sensors of a three-tank system. The previous approach to the robustness problem, based on a bank of UIFDO's, detected only accentuated faults (Wünnenberg, 1990). This fact was due to the inherent limitation of the mathematical model used in the design of that FDI system.

The main drawback is, however, that one requires data from faulty conditions. Those data can be collected either directly from the process, if available, or with the help of a simulation model that has to be as realistic as possible (Frank and Köppen-Seliger, 1997). The latter possibility is of special interest for collecting data corresponding to different faulty situations in order to assess the performance of the diagnosing system. Those data are not generally available from the real process. On the other hand, the experience gained using the ANN structures has evidenced the considerable computational effort that is implied in the design of the neural observers. Evolutionary algorithms of genetic type have been proposed to tackle different kinds of problems in the ANN research area (Man *et al.*, 1997). It is expected that those results would be applicable to the design of neural observers for fault diagnosis as well.

Acknowledgements

The first author gratefully acknowledges the support of this research by the Alexander von Humboldt Foundation, Germany. The second author would like to acknowledge the support for the completion of this work by the EC INCO-Copernicus project on 'Integration of Quantitative and Qualitative Fault Diagnosis Methods within the Framework of Industrial Application' (ERBIC–15 CT–97–0714).

References

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

Ayoubi M. (1996): *Nonlinear System Identification Based on Neural Networks with Locally Distributed Dynamics and Application to Technical Processes.* — Düsseldorf: VDI Verlag GmbH, Reihe 8, No.591.

Back A.D. and Tsoi A.C. (1991): *FIR and IIR synapses, a new neural network architecture for time series modeling.* — Neural Computation, Vol.3, No.3, pp.375–385.

Back A.D. and Tsoi A.C. (1992): *Nonlinear system identification using multilayer perceptrons with locally recurrent synaptic structure*, In: Neural Networks for Signal Processing II (S.Y. Kung, F. Fallside, J.Aa. Sorenson and C.A. Kamm, Eds.). — Piscataway, NJ: The IEEE Inc.

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

Demuth H. and Beale M. (1996): *Neural Network Toolbox.* — Natick, MA: The MathWorks Inc.

Frank P.M. (1994): *Enhancement of Robustness in Observer-Based Fault Detection.* — Int. J. Contr., Vol.59, No.4, pp.955–981.

Frank P.M. and Köppen-Seliger B. (1997): *New developments using AI in fault diagnosis.* — Engineering Applications of Artificial Intelligence, Vol.10, No.1, pp.3–14.

Frank P.M., Alcorta-Garcia E. and Köppen-Seliger B. (1997): *Modelling for fault detection and isolation.* — Prep. ESF Workshop *Control of Complex Systems*, Budapest, Hungary, pp.111–129.

Frank P.M., Schreier G. and Alcorta-Garcia E. (1999): *Nonlinear observers for fault detection and isolation*, In: New Directions in Nonlinear Observer Design (H. Nijmeijer and T.I. Fossen, Eds.). — Berlin: Springer Verlag.

Haykin S. (1994): *Neural Networks – A Comprehensive Foundation.* — New York: MacMillan College Publ. Comp.

Isermann R. and Ballé P. (1997): *Trends in the application of model-based fault detection and diagnosis of technical processes.* — Contr. Eng. Pract., Vol.5, No.5, pp.709–719.

Isermann R., Ernst E. and Nelles O. (1997): *Identification with dynamic neural networks — Architectures, comparisons, applications.* — Prep. IFAC Symp. *System Identification*, Fukuoka, Japan, pp.947–972.

Köppen-Seliger B. (1997): *Fehlerdiagnose mit künstlichen neuronalen Netzen.* — Düsseldorf: VDI Verlag GmbH, Reihe 8, No.632.

Liang F. and ElMaraghy H.A. (1993): *Multistep localized adaptive learning RBF networks for nonlinear system identification.* — Proc. ECCA/IFAC 2nd *European Control Conf.*, Groningen, The Netherlands, Vol.1, pp.111–116.

Man K.F., Tang K.S., Kwong S. and Halang W.A. (1997): *Genetic Algorithms for Control and Signal Processing.* — London: Springer Verlag.

Marcu T. (1996): *Pattern recognition techniques using fuzzily labeled data for process fault detection.* — J. Appl. Math. Comp. Sci., Vol.6, No.4, pp.815–840.

Marcu T. and Mirea L. (1997): *Robust detection and diagnosis of process faults using neural networks.* — IEEE Contr. Syst., Vol.17, No.5, pp.72–79.

Marcu T., Mirea L. and Mensler M. (1997): *Neural approaches to observer-based diagnosis of faults in dynamic systems.* – Prep. IFAC Symp. *System Identification*, Fukuoka, Japan, Vol.3, pp.1179–1184.

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

Sorsa T., Suontausta J. and Koivo H.N. (1993): *Fault diagnosis of dynamic systems using neural networks.* — Prep. *IFAC World Congress*, Sydney, Australia, Vol.7, pp.453–456.

Tsoi A.C. and Back A.D. (1994): *Locally recurrent globally feedforward networks: A critical review of architectures.* — IEEE Trans. Neural Networks, Vol.5, No.2, pp.229–239.

Williams R.J. and Zipser D. (1990): *Gradient-based learning algorithms for recurrent connectionist networks.* — Techn. Rep. NU-CCS-90-9, Northeastern University, College of Computer Science, Boston.

Wünnenberg J. (1990): *Observer-based fault detection in dynamic systems.* — Düsseldorf: VDI Verlag GmbH, Reihe 8, No.222.

Yokohama Y., Kohda T. and Inoue K. (1992): *Synthesis of optimal control using neural network with mixed structure.* — Prep. IFAC/IFIP/IMACS Int. Symp. *Artificial Intelligence in Real-Time Control*, Delft, The Netherlands, pp.289–294.

Received: 17 December 1998
Revised: 20 April 1999