Int. J. Appl. Math. and Comp. Sci., 1999, Vol.9, No.3, 471–518

ARTIFICIAL INTELLIGENCE APPROACHES TO FAULT DIAGNOSIS FOR DYNAMIC SYSTEMS

Ron J. PATTON*, Carlos J. LOPEZ-TORIBIO*
Faisel J. UPPAL*

Recent approaches to fault detection and isolation (FDI) for dynamic systems using methods of integrating quantitative and qualitative model information, based upon artificial intelligence (AI) techniques are surveyed. In this study, the use of AI methods is considered an important extension to the quantitative model-based approach for residual generation in FDI. When quantitative models are not readily available, a correctly trained artificial neural network (ANN) can be used as a non-linear dynamic model of the system. However, the neural network does not easily provide insight into model behaviour; the model is explicit rather than implicit in form. This main difficulty can be overcome using qualitative modelling or rule-based inference methods. For example, fuzzy logic can be used together with state-space models or neural networks to enhance FDI diagnostic reasoning capabilities. The paper discusses the properties of several methods of combining quantitative and qualitative system information and their practical value for fault diagnosis of real process systems.

Keywords: artificial intelligence methods, fault-diagnosis, residual generation, fuzzy modelling, neuro-fuzzy systems.

1. Introduction

There is an increasing demand for man-made dynamical systems to become safer and more reliable. These requirements extend beyond normally accepted safety-critical systems of nuclear reactors, chemical plants or aircraft, to new systems such as autonomous vehicles or fast rail systems. The early detection of faults can help avoid system shut-down, breakdown and even catastrophes involving human fatalities and material damage. A system which includes the capacity of detecting, isolating, identifying or classifying faults is called a *fault diagnosis* system. During the last two decades many investigations have been made using analytical approaches, based on quantitative models. The idea is to generate signals that reflect inconsistencies between nominal and faulty system operation. Such signals, termed residuals, are usually generated using analytical approaches, such as observers (Chen and Patton, 1999), parameter estimation (Isermann, 1994a) or parity equations (Gertler, 1998) based on

* Control and Intelligent Systems Research Group, School of Engineering, the University of Hull, Cottingham Road, Hull, HU6 7RX, UK, e-mail: r.j.patton@eng.hull.ac.uk.

analytical (or functional) redundancy. Considerable attention has been given to both research and application studies of real processes, using analytical redundancy as this is a powerful alternative to the use of repeated hardware (hardware or software redundancy).

The monitoring of faults in feedback control system components has come to be known as fault detection and isolation (FDI). The procedure of generating control action which has a low dependency on the presence of certain faults is known as fault-tolerant control. Figure 1 shows the general schematic arrangement appropriate to many fault-tolerant control systems with four main components: the plant itself (including sensors and actuators), the FDI unit, the feedback/feed-forward controller, and the supervision system.

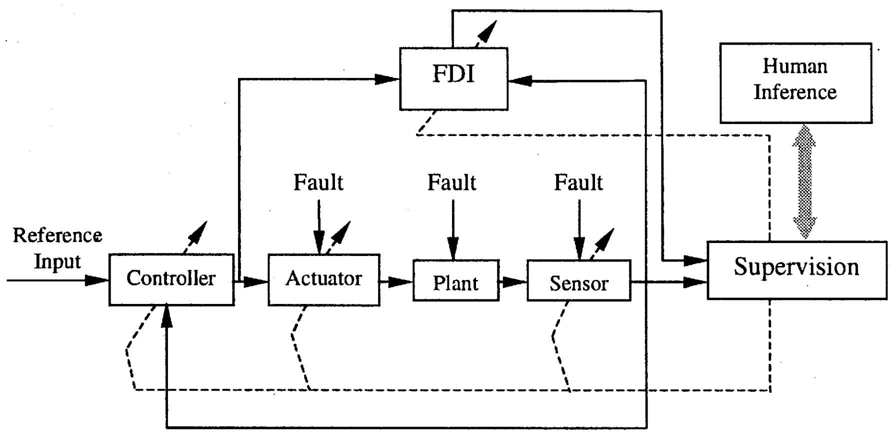

Fig. 1. Scheme of a fault-tolerant control system.

The solid line represents the signal flow, and the dashed line represents adaptation (tuning, scheduling, reconfiguration or restructure). The plant is considered to have potential faults in sensors, actuators (or other components). The FDI unit provides the supervision system with information about the onset, location and severity of any faults. Based on system inputs and outputs together with fault decision information from the FDI unit, the supervision system will reconfigure the sensor set and/or actuators to isolate the faults, and tune or adapt the controller to accommodate the fault effects.

Early detection and isolation of small, incipient (rather difficult to detect) faults can be achieved using model-based processing of all measured variables, using either qualitative or quantitative modelling. Neural networks and fuzzy logic techniques are now being investigated as powerful modelling and decision making tools, along with the more traditional use of non-linear and robust observers, parity space methods and hypothesis-testing theory.

Requirements for a precise and accurate analytical model imply that any resulting modelling error will affect the performance of the resulting FDI scheme. This is particularly true for non-linear systems, which represent the majority of real processes.

To circumvent this precision problem (at least in part), more abstract models, based on qualitative physics (de Kleer and Williams, 1987; Kuipers, 1994; Lunze and Schiller, 1999; Shen and Leitch, 1993) may be used. Alternatively, fuzzy-logic rules may be developed to either assist or replace the use of a model for diagnosis (Dexter, 1995). The key advantage of fuzzy logic is that it enables the system behaviour to be described by *if-then* relations. This can provide valuable information for the operator to understand the causes of faults. This is an attractive feature of fuzzy logic but it also has major limitations. In many applications the knowledge that describes the system behaviour is contained in data sets. The designer then has to derive *if-then* rules from the data sets manually, which imposes a major effort with large data sets.

Some research has been based upon neural networks which can be trained to reproduce a specified system behaviour from the data sets alone. Neural networks can, indeed, provide an excellent framework for dealing with non-linear systems (Leonard and Kramer, 1993; Naidu *et al.*, 1990). The main feature of neural networks is their ability to model any non-linear functions, given suitable weighting factors and an appropriate architecture. There has been a substantial body of work in recent years describing the use of neural networks for fault diagnosis of non-linear systems (Hennerberger *et al.*, 1993; Hoskins and Himmelblau, 1988; Lane *et al.*, 1992; Patton and Chen, 1996; Patton *et al.*, 1994; Wang *et al.*, 1994).

However, whilst such a configuration can be well trained on numerical data, heuristic knowledge from experts cannot easily be incorporated. It is also argued that, due to their 'black-box' characteristics, conventional neural networks do not give an insight into the behaviour of the system which is sufficiently comprehensible by the operator. Another drawback of substituting the operator's 'intelligence' by an automated analytical approach is that the operator's expertise, built up over several years, is simply not used. This is mainly due to the inability of analytical methods to represent symbolic information.

In the authors' opinion, a robust FDI system should combine both numerical (quantitative) and symbolic (qualitative) information. Some investigators tackled this problem by combining parameter estimation or observers with fuzzy logic (Frank and Kuipel, 1993; Isermann, 1994b). The main idea has been to generate residuals using either parameter estimation or observers, and allocate the decision-making to a fuzzy-logic inference engine. In so doing, it has been possible to include symbolic knowledge with the quantitative information and, thereby, minimise the false alarm rate. Indeed, the key benefit of fuzzy-logic is that it lets the operator describe the system behaviour or the fault-symptom relationship with simple *if-then* rules.

This paper gives an outline of AI methods which are considered a powerful extension to quantitative/analytical approaches to fault detection and isolation (FDI) for dynamic systems.

One approach is to use a fuzzy rule-base to select the dynamic model which is most appropriate for a particular operating point (Tanaka *et al.*, 1996; Wang *et al.*, 1995). This is the so-called fuzzy inference multiple-model approach. The idea has been borrowed from recent control research and applied to FDI problems by Lopez-Toribio *et al.* (1998).

In another approach, it is important to be able to structure a *quantitative model* in a way that qualitative knowledge about the process could be included as well as extracted. The underlying concept is to structure a neural network, which can model highly non-linear systems efficiently, in a fuzzy-logic format; the network could therefore be trained more rapidly and will also provide a linguistic description about the causes of faults. Expert knowledge could also be included in the same framework. The B-spline network can be a suitable network architecture for this problem due to an interesting equivalence relation with the function of fuzzy rule sets (Brown and Harris, 1994a). The difficulty with this approach is the rapidly increasing complexity of the rule base with the system order and complexity.

There are many neuro-fuzzy structures designed to combine the advantages of both neural networks and fuzzy logic. These structures have been successfully applied to a wide range of applications from industrial processes to financial systems, because of the ease of rule base design, linguistic modelling, application to complex and uncertain systems, inherent non-linear nature, learning abilities, parallel processing and fault tolerance abilities (Ayoubi, 1995). However, successful implementation depends heavily on prior knowledge of the system and the training data. There are three common methods of combining neural networks with fuzzy logic:

1. Fuzzification of the inputs or outputs of the neural networks.
2. Fuzzification of the interconnections of conventional neural networks.
3. Using neural networks in fuzzy models where neurons provide the necessary membership functions and rule base.

These approaches are either very complex or not adequate enough to provide approximation power and qualitative knowledge. Recent research focuses on neural networks, for example B-spline neural networks, which can be used to identify the process using a neural network architecture and also extract some qualitative knowledge of the system. This paper presents an application of B-spline neural networks to integrate the approximation techniques of neural networks and qualitative approach of fuzzy logic. The network is used to identify different parts of a sugar factory evaporation plant. The operator can also include any heuristic knowledge about the plant. Unlike many other neuro-fuzzy approaches, the B-spline network offers a simple and easy-to-build framework.

2. Principles of Model-Based Fault Diagnosis

The aim of a quantitative model-based fault diagnosis is to generate information about the location and timing of a fault in a system, using the measurements available in that system, as well as the *precise* mathematical relationships that relate them. Figure 2 illustrates the conceptual structure of a model-based fault diagnosis system, which comprises the following main stages.

Residual signal:

$$r(s) = H_u u(s) + H_y y(s) \tag{1}$$

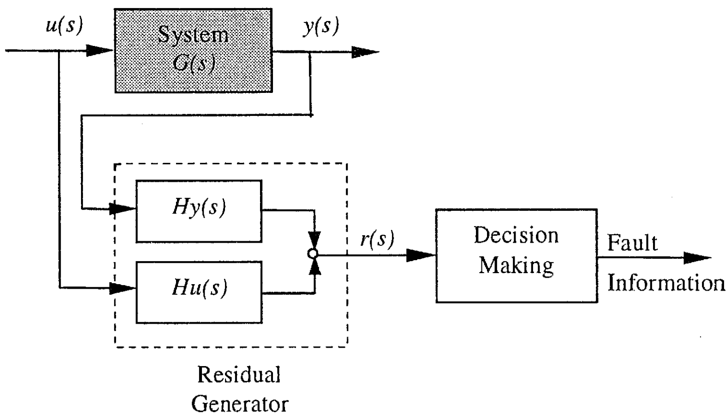

Fig. 2. A model-based fault diagnosis structure.

Objectives: Choose H_u and H_y so that

$$r(s) = 0 \text{ when no fault occurs}$$

$$r(s) \neq 0 \text{ when a fault occurs}$$

1. *Residual generation:* This is an algorithm which processes the measurable inputs and outputs of the system to generate the residual signal. It uses the model, describing the relationship between those variables in exact mathematical terms, and any inconsistency in this relationship will indicate a fault in the system. The residual must, therefore, be different from zero when a fault occurs and zero otherwise.
2. *Decision making:* The residuals are then examined for the likelihood of faults, and a decision rule is then applied to determine if any fault has occurred. The decision process may be based on a simple threshold test, on the instantaneous values or moving averages of the residuals, or it may consist of methods of statistical decision theory, e.g. likelihood ratio testing or sequential probability testing. The successful detection of a fault is followed by the fault isolation procedure whose aim is to locate the fault. A single residual signal is sufficient to detect the occurrence of a fault but a set of residuals is required for fault isolation. From the various approaches used for the design of model-based residual generators, three separate classes can be identified:
   - *Observers*: The underlying idea is to estimate the system outputs from the available inputs and outputs of that system (Patton, 1997). The residual will then be a weighted difference between the estimated and actual outputs. The flexibility in selecting the observer gain has been fully exploited in the observer, yielding a rich variety of fault detection schemes.
   - *Parity relations*: They are based either on a technique of direct redundancy, making use of static algebraic relations between sensor and actuator signals

or alternatively, upon temporal redundancy, when dynamic relations between inputs and outputs are used.

- *Parameter estimation*: This approach makes use of the fact that component faults of a dynamic system can be thought of as reflected in the physical parameters as e.g. friction or mass velocity resistance. It detects faults through the estimation or identification of model parameters, using non-parametric models.

The main assumption made when using the above methods is that a precise mathematical model of the plant is required. This makes quantitative model-based approaches very difficult to use in real systems, since any unmodelled dynamics can affect the performance of the FDI scheme. A way to overcome this is to design robust algorithms where the effects of disturbances on the residual are minimised and the sensitivity to faults maximised. A large number of approaches had been developed including unknown input observers and eigenstructure assignment observers (Chen and Patton, 1999), frequency domain techniques for robust FDI filters such as H_∞ (Edelmeyer *et al.*, 1994; 1997) and the minimisation of multi-objective functions (Chen *et al.*, 1997a).

To isolate faults, the residual signal has to be classified further, to indicate which system component has failed. One commonly accepted approach to fault isolation is to generate a set of structured signals. The aim is to have each residual sensitive to certain groups of faults and insensitive to others (Chen *et al.* 1996). The relationship between faults and residuals, however, can be non-linear such as with multiplicative faults, thereby making the fault very difficult to isolate.

3. Fault Diagnosis via Neural Networks

To overcome some of the difficulties of using mathematical models and make FDI algorithms more applicable to real systems, neural networks can be used both to generate residuals and to isolate a fault (Chen and Patton, 1999). A neural network is a processing system that consists of a number of highly interconnected units called neurons. A single neuron is very simple in construction, but a number of neurons connected together in a highly parallel way give high processing power. The neurons are interconnected by a large number of 'weighted links'. Each neuron can be considered as a mathematical function that maps the input and output spaces with several inputs. The inputs are connected to either the inputs of the system or the outputs of other neurons in the system. The output of one neuron affects the outputs of other neurons and all the neurons connected together can perform complex processes. There are a number of neural network architectures with different types of neurons and connections. Artificial neural networks are inspired from the investigations of neuro-biologists, psychologists and physicists, who have been studying the function of the human brain for several centuries. Details of the development of artificial neural networks can be found in many texts (Hagan *et al.*, 1996; Haykin, 1994). Some of the most common architectures used for fault diagnosis are discussed in later sections.

Indeed, one of the main features of neural networks is their ability to learn from examples. Hence, neural networks can be trained to represent relationships between past values of residual data (generated by another neural network) and those identified with some known fault conditions. The configuration used by Chen and Patton (1999) involved a multi-layer feed-forward network configuration. Whilst such a configuration can be extremely well trained on numerical data once the output is known, symbolic knowledge from experts cannot easily be incorporated.

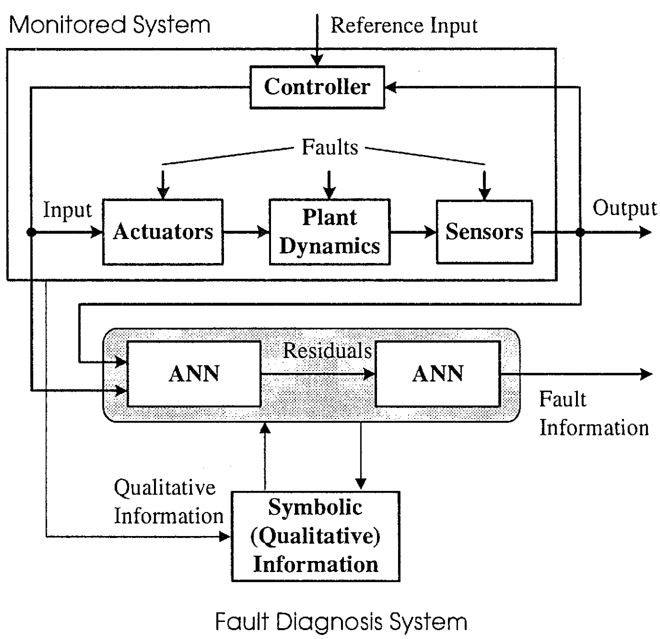

Fig. 3. Neural networks scheme for FDI.

The approximation abilities of neural networks show a great promise in non-linear control systems as they can approximate any non-linear function, given suitable weighting factors and architecture. Traditional methods for dealing with non-linear systems depend on generating a linear model of the system at some operating point. No linearisation is required for the neural networks, although the mathematical model used in traditional methods is very sensitive to modelling errors, parameter variation, noise and disturbance. In fact, no mathematical model of the system is needed to implement a neural network. On-line training makes it possible to change the FDI system easily in the cases where changes are made in the physical process, control system or parameters. A suitably trained neural network can *generalise* when presented with inputs not appearing in the training data. Neural networks have the ability to make intelligent decisions in the cases of noisy or corrupted data. They also have a highly parallel structure which is expected to achieve a higher degree of fault-tolerance than conventional schemes (Hunt *et al.*, 1992). Neural networks can operate simultaneously on qualitative and quantitative data and they are readily applicable to multivariable systems. Neural networks use a 'black-box approach', which can be suitable for FDI of complex systems, where the internal knowledge is not fully known. Neural networks can also be applied for process condition monitoring, where the focus is on small irreversible changes in the process which develop into bigger faults. Yin (1993) demonstrated the application of MLP and Kohonen self-organising feature map (KFM) to the predictive maintenance or condition-based maintenance of electrical drives, particularly induction motors. The first method utilises supervised learning and the other unsupervised.

Application studies. Neural networks have been successfully employed in many applications, including fault diagnosis of non-linear dynamic systems (Dong and McAvoy, 1996; Wang *et al.*, 1994). Multi-layer perceptron (MLP) networks are used to detect leakages in electro-hydraulic cylinder drive in a fluid power system (Stewart, 1995; Watton and Pham, 1997). Ilott and Griffiths (1997) applied artificial neural networks to the fault diagnosis of pumping machinery. They showed that maintenance information can be obtained from the monitored data using a neural network instead of a human operator. Crowther *et al.* (1998), applied a neural network to fault diagnosis of hydraulic actuators. They showed that experimental faults can be diagnosed using neural networks trained only on simulation data. Neural networks are used to detect an internal leakage in the control valves and motor faults in process plants (Sharif and Grosvenor, 1998). They have also been applied to the problem of joint faults in robots, using pattern recognition. The joint-backlash of robots is diagnosed by monitoring its vibration response during normal operation (Pan *et al.*, 1998). James and Yu (1995) used a neural network for the condition monitoring and fault diagnosis of a high-pressure air compressor valve. The neural network-based FDI scheme can also appear when further increases in fault levels might be likely, thus giving the operator time to take necessary action (Boucherma, 1995).

Neural networks have been found to give more information with regard to multiple-fault conditions than some other methods (steady-state position error, time series analysis). Dynamical neural networks are applied to on-line fault detection of power systems, particularly dynamic feed-forward networks, time delay and modified Elman neural networks (Carley, 1997). Neural networks have also been considered for fault detection, isolation and reconfiguration (FDIR) of aircraft systems, which are highly non-linear complex and require a high performance (Chiang, 1996). Neural network-based fault diagnosis and process-monitoring methods can be applied to safety critical and highly non-linear processes such as chemical plants (Calado and Sa da Costa, 1999) and nuclear reactors. Marseguerra *et al.* (1996) demonstrated an application of a neural network to the fault detection in a pressurised water reactor pressuriser.

Strategies for fault diagnosis. It is clear that neural networks can be applied to fault diagnosis in two different approaches: *the pattern recognition* approach and *residual generation-decision making.* The second approach is generally more suitable for dynamic systems and comprises *residual generation* and *decision making* stages in just the manner outlined in Section 2. In the first stage, a residual vector r is determined in order to characterise each fault. Ideally, neural network models identify all classes of system behaviour. The second stage, *decision making*, or *classification*, processes the residual vector r to determine the location and occurrence time of the faults. A neural network can be used for classification in conjunction with other residual generating methods, e.g. non-linear observers.

Taxonomy of neural networks. There is a large number of neural network architectures. Some of the most frequently used structures for fault diagnosis are listed in Fig. 4. Other networks used in recent applications include dynamic backpropagation networks (Narendra, 1998), Adaptive Resonance Theory (ART) networks (Lin and Wang, 1993) and Cerebellar Model Articulation Controller (CMAC) networks (Brown and Harris, 1994b; Leonhardt *et al.*, 1995). Each of these architectures offers different characteristics to suit different applications. Recent research focuses on networks which can optimise their structure during training. Ren and Chen (1999) proposed a new type of neural network in which the dynamical error feedback is used to modify the inputs of the network. Xiong *et al.* (1999) proposed multiple neural networks cascaded together instead of a single optimal network, for improved predictions. Tan and Saif (1999) proposed a new method called the *dynamic gain matrix technique* used with a neural network to isolate the sources of sensor faults. These can be located by comparing the dynamic gains of the system model in healthy and faulty situations.

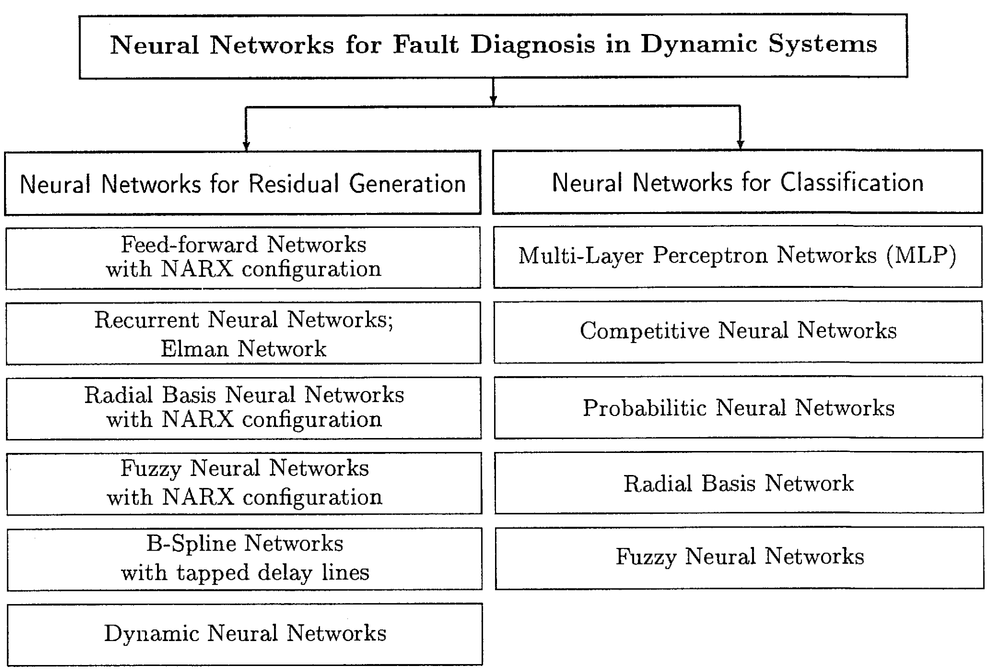

Fig. 4. Neural network architectures.

There are two basic applications of neural networks to fault diagnosis, e.g. modelling and classification. Modelling includes identification of the system under normal and/or different operating conditions. As outlined above, this capability is used to generate residuals comparing the model states or outputs with those of the actual system. Typical architectures chosen for this purpose are feed-forward networks, Radial Basis Function (RBF) networks and dynamic neural networks because of their powerful approximation and generalising abilities. Many researchers concentrate on the identification or modelling capabilities of neural networks (Brown and Harris, 1994c; Hunt *et al.*, 1992; Levin and Narendra, 1996).

The second useful function is classification which is used to detect and isolate each fault. Given suitable training data for all fault conditions, different neural architectures can be trained to classify each fault. This can be achieved in two ways. The first approach, which follows the traditional FDI scheme consists in modelling the system, generating residuals using a neural network and using a second neural network to classify each fault. Chen and Patton (1999) also showed that a single neural network is capable of performing both the tasks simultaneously, with increased training time and complexity. Fault isolation requires that the training data be available for all expected faults in terms of residual values or system measurements.

For identification of dynamic systems, neural networks also need to have some dynamics involved in the structure (Hunt *et al.*, 1992). Dynamic or recurrent networks have dynamic elements within their structure. For networks without internal dynamic elements the most common way of identification is to use tapped delay lines such as Non-linear Auto Regressive eXogenous training structure (NARX model), (Billings and Leontaritis, 1985; Doherty *et al.*, 1994) for its ease of implementation. Narendra (1998) discussed different approximate models for non-linear systems, which are suitable for different applications. The basis of the NARX model is the assumption that the non-linear system dynamics can be represented by the equation

$$y_{NN}(t) = f\Big(u(t-k),\dots,u(t-k-n_u+1),y(t-1),\dots,y(t-n_y)\Big) + e(t) \quad (2)$$

where y is the process output, u is the process input, u is the error, f is the non-linear function represented by the neural network, $n_u = n_y = n$ is the order of the model and $k \geq 1$ is the *model dead-time.*

3.1. Feed-Forward Neural Networks with Back-Propagation

Feed-forward neural networks have a simple architecture and can be used for both residual generation and classification. Feed-forward networks as shown in Fig. 5 are used for *identification* or *residual generation* because they do not have internal dynamic elements. This type of simple network is not efficient when modelling non-linear dynamic systems, but the ease of implementation makes it ideal for the cases where the transient error is less important. Each neuron outputs only to the neurons of the next layer (Caudill and Butler, 1992). The input layer acts as a fan-out of the input signal pattern. The hidden layer neurons act as feature detectors which encode in their weights the features present in the input patterns. These features are then used by the output layer to determine the correct output. More than one hidden layer is possible depending on the complexity of the system.

The net input for each of the neurons in each layer is calculated by $I_i = \sum_{j=1}^{m} w_i X_j$, where w_i is the weight connecting the i-th and the j-th neuron and

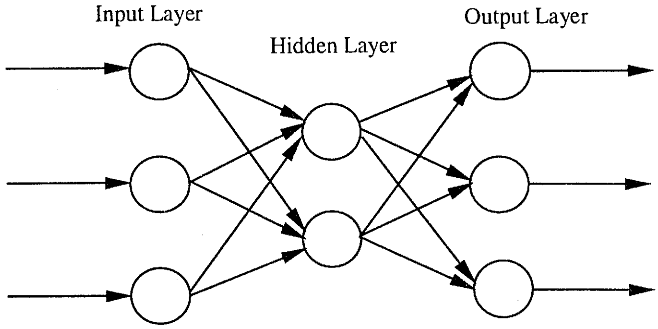

Fig. 5. Feed-forward neural networks.

x_j is the output of the j-th neuron of the previous layer. The sigmoid function is most commonly used as the activation function, i.e.

$$y = f(I_i) = \frac{1}{1+e^{-I_i}} \qquad \text{for } i = 1, \ldots, n \tag{3}$$

To train the network, input patterns are applied at the inputs in turn, the error of the output layer neurons is calculated directly by comparing these with the desired outputs and is back-propagated from the output layer and multiplied by the derivative of the middle-layer neurons activity to determine the error for the previous layer neurons. The weight change equation is given by

$$\Delta w_{ij} = \beta E f(I) + \alpha \Delta w_{ij}^{\text{previous}} \tag{4}$$

where β is the learning coefficient, E is the error for that particular neuron and α is the momentum coefficient.

The error E is determined by the following equations:

$$E_j^{\text{output}} = y_j^{\text{desired}} - y_j^{\text{actual}} \tag{5}$$

$$E_j^{\text{first/hidden}} = \frac{\mathrm{d}f(I_i^{\text{middle}})}{\mathrm{d}I} \sum_{j=1}^{n} w_{ij} E_j^{\text{next}} \tag{6}$$

For each pattern applied the weight is updated and the whole set of patterns is repeated until the network reaches a desired 'sum-square-error goal.' Simple feed-forward neural networks cannot be applied to dynamic problems involved in the fault diagnosis because they are basically static networks.

3.2. Recurrent Networks

Recurrent neural networks (Fig. 6) can model dynamic systems much better and they do not require *tapped delay lines*, as is the case of feed-forward networks, and employ much more computational power for training and implementation because past values of the signals within the network are also used. However, because of

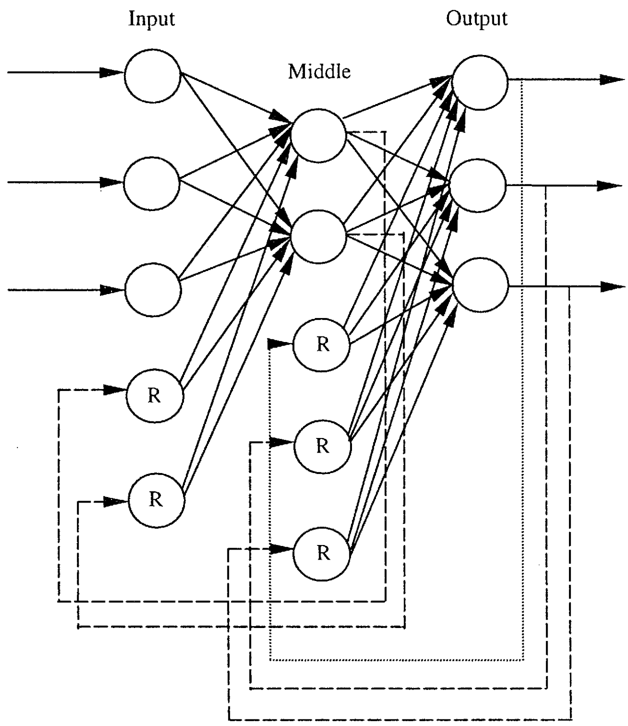

Fig. 6. Recurrent networks.

greater computational requirements, these are not usually used for the classification purpose. In recurrent networks (Caudill and Butler, 1992) each input activity pattern passes through the network more than once before it generates an output.

Recurrent back-propagation networks are the most commonly used networks in which recurrent neurons are added in to handle the feedback from the next layers neurons as shown in Fig. 6. The weights on the connections leading to recurrent neurons are usually fixed to unity. The activity equation includes the previous activity. Similarly, the weight change equation also depends on the previous activity and weights, thus this type of training is also called *back-propagation through time.* The following equations compute the activity for each neuron from time $t = 1$ to $t = N$:

$$I(t)_j^{\mathrm{mid}} = \sum_{i=1}^{3} w_{ij}^{\mathrm{in-mid}} y(t)_i^{\mathrm{in}} + \sum_{k=1}^{2} w_{kj}^{\mathrm{mid-mid}} y(t-1)_k^{\mathrm{mid}} \tag{7}$$

$$I(t)_j^{\mathrm{out}} = \sum_{i=1}^{2} w_{ij}^{\mathrm{mid-out}} y(t)_i^{\mathrm{mid}} + \sum_{k=1}^{3} w_{kj}^{\mathrm{out-out}} y(t-1)_k^{\mathrm{out}} \tag{8}$$

Starting with the error at $t = N$, the backward error is propagated for each layer from $t = N$ to $t = 1$:

$$E(N)_j^{\mathrm{out}} = \left(y(N)_j^{\mathrm{desired}} - y(N)_j^{\mathrm{actual}}\right) \frac{\mathrm{d}f(I(N))}{\mathrm{d}I} \tag{9}$$

$$E(N)_j^{\mathrm{mid}} = \left(\sum_{i=1}^{3} w_{ij}^{\mathrm{mid-out}} E(N)_i^{\mathrm{out}}\right) \frac{\mathrm{d}f(I(N)_j^{\mathrm{mid}})}{\mathrm{d}I} \tag{10}$$

The backward error propagation for all times prior to $t = N$ starts with

$$E(t)_j^{\mathrm{out}} = \left(\varepsilon(t) + \sum_{i=1}^{3} w_{ij}^{\mathrm{out-out}} E(t+1)_i^{\mathrm{out}}\right) \frac{\mathrm{d}f(I(t)_j)}{\mathrm{d}I} \tag{11}$$

where $\varepsilon(N)_j^{\mathrm{out}} = (y(N)_j^{\mathrm{desired}} - y(N)_j^{\mathrm{actual}})$, and continues with

$$E(t)_j^{\mathrm{mid}} = \left(\sum_{i=1}^{3} w_{ij}^{\mathrm{mid-out}} E(t)_i^{\mathrm{out}} + \sum_{k=1}^{1} w_{kj}^{\mathrm{mid-mid}} E(t+1)_k^{\mathrm{mid}}\right) \frac{\mathrm{d}f(I(t)_j^{\mathrm{mid}})}{\mathrm{d}I} \tag{12}$$

The weight change law for non-recurrent weights is

$$\Delta w(t)_{ij}^{\mathrm{in-mid/in-out}} = \beta E(t)_i y(t)_j \tag{13}$$

whereas the weight change law for recurrent weights is

$$\Delta w(t)_{ij}^{\mathrm{mid-mid/out-out}} = \beta E(t)_i y(t-1)_j \tag{14}$$

Catfolis (1996) pointed out that recurrent neural networks have the drawback that temporal knowledge about the process is difficult to use. The RTRL (Real Time Recurrent Learning) algorithm is suggested so that the temporal knowledge can be used in network training. Moreover, this results in reduced training times for RTRL networks.

3.3. Dynamic Neural Networks

Hunt *et al.* (1992) suggested a general structure for implementing a number of both dynamic and non-dynamic neural networks. The dynamic neural networks utilise neurons with dynamic elements as shown in Fig. 7. The dynamic elements give the network power to represent dynamic systems, when used for identification or residual generation. This type of network is more efficient than recurrent networks, especially when the number of inputs and outputs is large. The greater computational requirements mean that these networks are not generally used for classification. Recent research considers a wider class of dynamic networks both for identification and for diagnosis (Korbicz *et al.*, 1998; Patan and Korbicz, 1996).

The weighted summation gives the output

$$v_i(t) = \sum_{j=1}^{N} a_{ij} y_j(t) + \sum_{k=1}^{M} b_{ik} u_k(t) + w_i \tag{15}$$

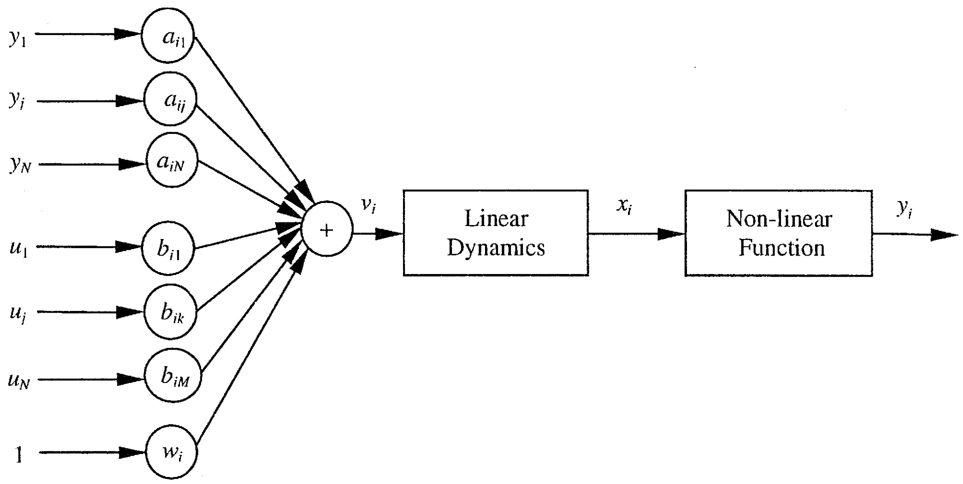

Fig. 7. General structure for dynamic neural networks.

where y_I are the outputs of all neurons, u_k are the external inputs, w_I are biasing factors and a_{ij}, b_{ik} are weighting coefficients. The linear dynamics can be described by the function $X_I(s) = H(s)V_I(s)$, where $X(s)$ is the Laplace transform of $x(t)$, $V(s)$ is the Laplace transform of $v(t)$ and $H(s)$ is the transfer function. In the time domain the above equation becomes $x_i(t) = \int_{-\infty}^{t} h(t-t')v_i(t')\,\mathrm{d}t'$. A non-dynamic non-linear function can be represented by $y_i = g(x_i)$ where g is a function upon x. To train such a network, the *dynamic backpropagation* method is commonly used.

3.4. Radial Basis Function (RBF) Networks

Radial basis function networks are single-hidden-layer feed-forward networks. They are concerned with the data clusters rather than data boundaries as in multi-layer perceptron networks, as discussed by Wilson (1998). For each input the distance between the input vector and vector of centres is calculated rather than the input itself and passed through the neuron activation function. RBF networks (see Fig. 8) are capable of approximating any function with arbitrary accuracy.

An RBF network can be represented by $y = \sum_{i=1}^{N} w_i f(||x - c_i||)$ where x is the input vector and c is the vector of centres. The Gaussian function of the form

$$f(u) = \exp\left[-\frac{u^2}{2\sigma^2}\right] \tag{16}$$

where σ denotes a smoothing factor (width), is the most commonly used although different non-linear functions are also applied in practice. The specific type of non-linear function is not very important to the performance of an RBF network (Narendra, 1998).

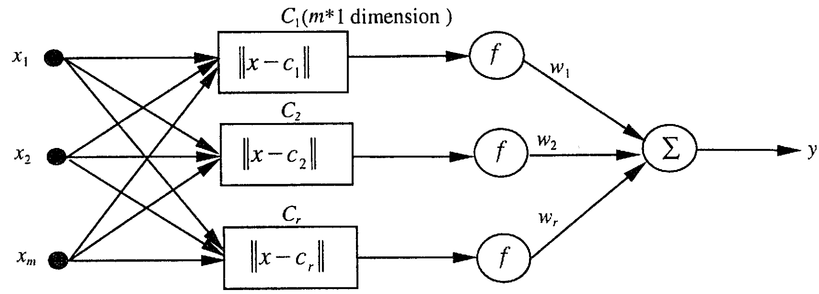

Fig. 8. Structure of RBF networks.

Temporal instability of MLP networks can be avoided without the excessive parametrisation required by B-spline networks (Wilson, 1998). The network is trained to minimise the cost function which is the *sum-squared-error*

$$J = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{17}$$

A typical network with m inputs and r neurons has the following parameters to optimise: r weights $(w_1, \ldots, w_r)$, Rm centres and r widths. If weights and widths are pre-allocated, then the optimisation of weights constitutes a linear least-squares problem. Hence, a global minimum can be found with relatively little computational effort. This gives a huge advantage over MLPs in terms of the training time and avoidance of local minima. *On-line training* is possible if recursive least-squares (RLS) algorithms are adopted. Pre-determined parameters reduce the number of degrees of freedom available to the network. The following methods of training can be applied for RBF networks:

1. The *centres* are distributed randomly across the input space with a single width value. The network is quick to train and the results fit reasonably well, but the output function is not smooth. If the centres are distributed uniformly, they will be forced into regions with few or no data points. This is a less efficient method and to work reasonably, a large number of neurons is required. Moreover, the results can be improved by selecting the centres as a random subset of the input space.
2. The centres are distributed such that their distribution mirrors that of the training data, which implies implementation of a type of vector quantisation or a clustering algorithm, *K-means clustering.* (Moody and Darken, 1989; Wilson, 1998)

RBF networks overcome some of the disadvantages of MLPs by using non-monotonic transfer functions based on the Gaussian density function (Dalmi *et al.*, 1999). Moreover, the RBF network produces more robust decision surfaces and the training is faster than for the MLP.

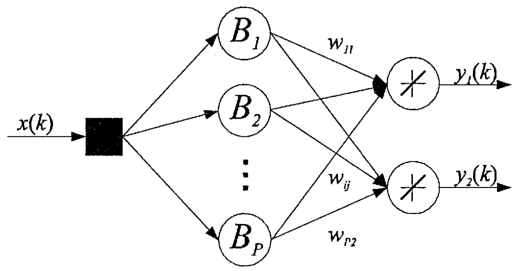

Fig. 9. Architecture of a B-spline network with one input and two outputs.

3.5. B-Spline Neural Network

A B-spline neural network is another single-hidden-layer neural network similar to the RBF network, using B-spline basis functions and can be divided into the following stages:

- A space of all possible inputs, which comprises the measurable inputs and outputs of the system being monitored.
- Basis functions: they are associative cells defined in the input space and joined at breakpoints, referred to as knots. These functions perform a non-linear transformation of the input into the interval $[0, 1]$. Their shape, size and overlap determine the modelling capabilities of the resulting network.
- Weight vector ω_i: these are linear coefficients which are adjusted in the training phase of the algorithm.

The structure of B-spline neural network is shown in Fig. 9. It is defined by its order n, number of B-spline functions p, weight matrix w and the normalised space of possible inputs. Parameters n and p define the basis functions which specify the shape, size and overlap of the functions. The weight matrix w consists of linear coefficients, adjusted during the learning process. These parameters determine the modelling capabilities of the network (Ahlberg *et al.*, 1967; Sard and Weintraub, 1971).

The i-th output of a network can be represented as a linear combination of its basis function $B_{n,j}(x)$:

$$y_i(k) = \sum_{j=1}^{P} w_{ij} B_{nj}(x) \tag{18}$$

where n denotes the order of the basis functions, w_{ji} (weight) is the parameter associated with the j-th basis function and the i-th output, and P is the number of B-spline functions for each input. To construct B-spline functions of any order, the following recurrent relationship can be used:

$$B_{n,j}(x) = \left(\frac{x\lambda_{j-n}}{\lambda_{j-1} - \lambda_{j-n}}\right) B_{n-1,j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-n+1}}\right) B_{n-1,j}(x) \tag{19}$$

where

$$B_{1,j}(x) = \begin{cases} 1 & \text{if } x \in I_j \\ 0 & \text{otherwise} \end{cases}$$

and j is the B-spline index ascribed to the region of local support $\lambda_{(j-n)} \le x \le \lambda_{(j)}$. Figure 10 shows different orders and partitions for B-spline functions.

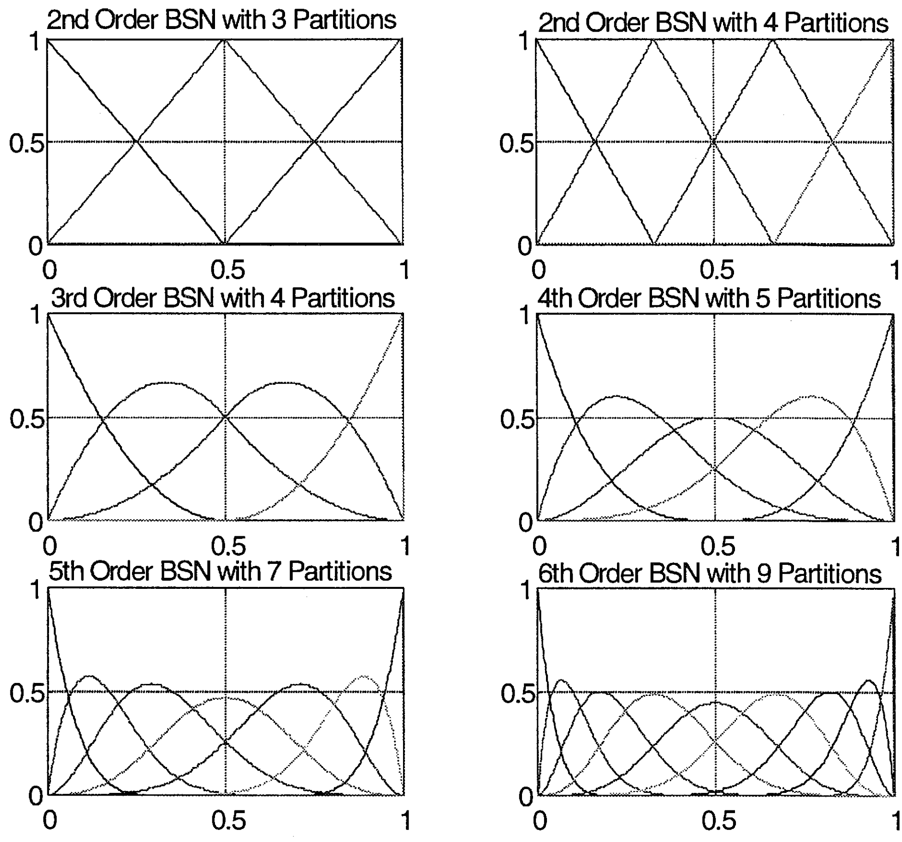

Fig. 10. Different orders and numbers of B-spline functions.

After selecting the B-spline function parameters, the training of the network consists in finding weight coefficients ω_i that minimise the cost function,

$$J + \frac{1}{N} \sum_{i=1}^{N} \left(\hat{r}(t) - r(t)\right)^2 \tag{20}$$

where N denotes the number of training sets, $r(t)$ stands for the target signal and $\hat{r}(t)$ is the network's outputs.

To find an optimal set of weights, different methods can be used including off-line least-squares, on-line recursive least-squares, and back-propagation of feed-forward networks.

Multi-dimensional B-spline models are used for multiple-input multiple-output systems. The equivalent basis functions are determined as a product of one-dimensional models,

$$B(x) = \prod_{i=1}^{p} B_p(x_p) \tag{21}$$

3.6. Design Issues of Applying Neural Networks for Fault Diagnosis

1. As discussed earlier, neural networks present a 'black box' which does not show the rules governing the operation. This does not enable the user to understand the system and predict its behaviour in uncertain situations. As discussed in the following sections, B-spline neural networks can be used not only for identification and classification, but also to extract and include some heuristic knowledge about the system.
2. The *training time* required for a specific application and the complexity of the training algorithm present another limitation. The earlier back-propagation algorithm used to train MLPs takes a long time to train and is generally an off-line method for training. RBF networks are capable of on-line adaptive training if required (Wilson, 1998) but use a large number of neurons if the I/O space is large. B-spline networks are suitable for systems with a smaller number of variables because the computational effort required to train such networks is an exponential function of the number of inputs to the network. To accelerate convergence, state variables with additional terms can be used in training (Watton and Pham, 1997).
3. Neural networks tend to approximate the exact training data including noise, if a more complex network architecture is used than the original system. This is called an 'over-parametrised network.' Similarly, if the network architecture chosen is not adequate enough, it will 'under-fit' the system behaviour. This is called an 'under parametrised network.' Over-trained networks and under-trained networks also show the same phenomena. To overcome these difficulties, training data are filtered for noise and the training parameters such as 'sum-squared-error' and 'number of epochs' are selected carefully. James and Yu (1995) showed that algorithms can be developed which do not require initial guesses of weights and the number of neurons in the hidden layer. Keenan (1998) proposed artificial evolution of the neural network to eliminate the trial and error usually associated with ANNs. The evolution procedure produces varying objects derived from a base class, out of which better objects are selected on some predefined performance index. These methods have many limitations for the time being.
4. Neural networks which use neurons as membership functions, e.g. RBF and B-spline networks, do not generalise well when presented with data outside the training I/O space. For example, B-spline networks will generate the zero output if

the data is presented outside the I/O space. Other neural network architectures as MLP and fuzzy-logic based systems tend to generalise in a better way. On-line training should be used to update such networks (Wilson, 1998).

5. If some unknown fault conditions appear, the neural classifier is no longer valid because it is not trained to classify this type of fault. Adaptive training algorithms should be used with systems requiring on-line training. However, on-line training must be done carefully, otherwise the network might learn faulty instead of healthy system behaviour. The network will no longer be able to classify the fault.

6. Neural networks are often not suitable for real-world application problems because of their complexity. Many other techniques have been combined with neural networks, including fuzzy logic, genetic algorithms, adaptive modelling, etc. These combinations enhance the power of the neural network for solving complex problems.

7. Neural-network-based FDI methods usually require pre-processing or signal conditioning algorithms to reduce the effect of noise and disturbance and to enhance the fault features. Franckin *et al.* (1999) proposed a fault detection scheme using a neural network with fuzzy input pre-processor. The fuzzification of the inputs to the network enriches the information, making it easier for the network to diagnose and isolate faults.

8. It is not usually possible to acquire all the faulty data for neural network training. It might be very dangerous or impossible to acquire faulty data from the process. Thus *unsupervised training* is necessary in order to classify the faults not known *a priori*. There are two basic types of unsupervised learning schemes for neural networks: the Kohonen network and the Counter-Propagation Network (CPN) (Dalmi *et al.*, 1999).

4. Fault Diagnosis via Fuzzy Logic

Since 1965 when Zadeh introduced the theory of the fuzzy sets manipulating data that were not precise but rather 'fuzzy' and since the work of Mamdani (1974), industrial application studies using fuzzy logic controllers have reached a major position in systems engineering. Application areas include the process industry, electromechanical systems, traffic and avionics control and biomedical systems.

Fuzzy systems are useful in any situation in which the measurements taken are imprecise or their interpretation depends strongly on the context or on human opinion. The architecture for the knowledge-based model of a controller consists of three blocks (see Fig. 11). The fuzzification interface maps the input value to a suitable domain and converts it into a linguistic term or into a fuzzy set. The knowledge base contains information about the boundaries, possible transformations of the domains, and the fuzzy sets with their corresponding linguistic terms. This information represents the data base. In addition, the knowledge base contains a rule base consisting of linguistic control rules. The decision logic represents the processing unit which determines the corresponding output value from the measured input according to the knowledge

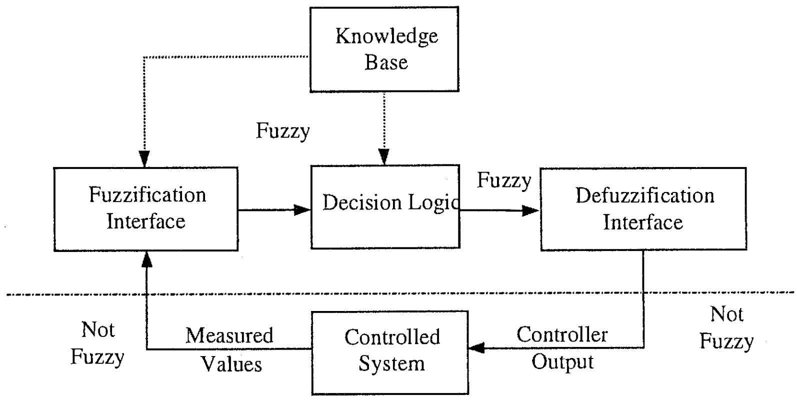

Fig. 11. Knowledge-based model of a controller.

base and the de-fuzzification interface determines a crisp output and carries out a transformation of the output value into the appropriate domain (Kruse *et al.*, 1994).

Although fuzzy systems theory is often applied to an industrial process, the applications often do not work well. Sometimes fuzzy logic designs are completed without mathematical rigour. The main tasks of finding appropriate membership functions and fuzzy rules are often determined simply by 'trial and error.' The rules can be obtained by means of optimisation methods. A fuzzy logic controller can be used to optimise and solve classical problems of PID controller tuning (Ruano *et al.*, 1999). Alternatively, when information from the system is not available, the fuzzy rules can then be optimised. Hu *et al.* (1999) show how fuzzy inference can be used for tuning controllers by evaluating and examining the functional behaviour of fuzzy PID-like controllers. The introduction of a convex optimisation method using LMI (Linear Matrix Inequalities) has facilitated the mixed design of robust fuzzy control and optimal fuzzy control for non-linear systems (Tanaka *et al.*, 1999). LMI optimisation has been used in order to design an optimal Takagi-Sugeno (T-S) observer based on a relaxed stability condition (Patton *et al.*, 1998). Another main approach to obtain the number, position and type of rules is to apply adaptive and learning algorithms to fuzzy systems or to apply neural networks to learn the parameters of the fuzzy system. For control and classification, neuro-fuzzy techniques are now receiving wide attention. Neuro-fuzzy methods can facilitate the development of fuzzy systems based on heuristic learning strategies obtained from neural network theory. Typical neuro-fuzzy approaches are NEFPROX, NEFCON (Nürnberger *et al.*, 1997), NEFCLASS (Nauck and Kruse, 1998) and ANFIS which combines back-propagation learning and least-mean-squares estimation (Jang and Sun, 1993).

4.1. Fuzzy Decision-Making

Fuzzy decision-making procedures are most useful when the information available is subjective, unclear, vague, or imprecise in some other way. Process control is a decision problem where a number of actions are available to the controller but only a set of selected control actions must be performed to achieve the specified control. The advantage of the fuzzy approach is that it supports, in a natural way, the direct integration of the human operator into the fault detection and supervision process. By avoiding an incorrect decision that can cause false alarms, the aim of the FDI decision making (for fault diagnosis) is to decide whether and where the fault in the system has occurred (Kuipel and Frank (1997). Fuzzy decision-making objectives are very similar to expert systems and supervisory control. Expert systems are used to simulate the problem-solving and decision-making processes of a human expert within a relatively narrow domain. This is done using special computer packages along with knowledge, information and databases (Ford, 1991; Tzafestas, 1989).

Formulation of decision-making. A decision can be formulated by a set of variables (sets, relations and functions) termed a quintuple $(S,\ st,\ C,\ m,\ dc)$ (Kaymak, 1998; Verbruggen and Babuška, 1999). By using available information, S stands for the possible actions where a selection of this set is performed. Furthermore, st is the set of uncontrolled variables but they must be included in the decision making process. C signifies the set of consequences, which must be included into a multi-criteria decision-making. Uncertainties resulting from the identification procedure and inherent uncertainties of the system are included, in part, in the consequences. Moreover, m is the relation used to obtain the decision-making solutions by mapping the space $S \times st$ into the set consequences as $S \times st \to C$. The decision-maker has, *a priori*, aims and objectives in a preference ordering. They are taken into account in dc as a decision function $dc : C \to \mathbb{R}$. A number of fuzzy decision-making methods for control has been applied for more than two decades, for example the formulation by Bellman and Zadeh (1970). For this approach, there is no distinction between the aims and constraints; both are included in the membership functions. Another application of decision-making for control is the *supervisory control* described by Sousa *et al.* (1998). This takes the form of a higher-level controller using a decision-making approach for the adaptation process of the available control actions.

Fuzzy sets have been applied to model-based predictive control to obtain a sequence of future control actions through cost function minimisation. To simplify the optimisation method, the *receding horizon principle* is applied because only the first control action is important and this is therefore the action applied by the controller to the process. In order to relate the objectives and constraints in predictive control, a fuzzy decision-making strategy is applied using multi-criteria (Babuška, 1997; Braake *et al.*, 1994; Kaymak, 1998; Sousa, 1998).

4.2. Fuzzy Clustering and Fuzzy Identification

For linear systems, identification techniques are well developed and established and are therefore widely applied. However, for non-linear systems less attention has been paid to identification procedures due to the complexities involved. To identify complex nonlinear systems it is common to obtain partitions of the available data, and then each partition or subset is approximated by a simple model. The data can be quantitative or qualitative information or a mixture. Clustering algorithms are not only used for classification and pattern recognition to construct fuzzy models but also for the simplification and optimisation in modelling.

Isoc (1998) used quasi-linear fuzzy models based on the Sugeno approach (from experimental measurement data according to the Box-Jenkins data sets). These were compared with the real system data sets and then with models obtained using other identification techniques. Various identification techniques to develop fuzzy models were used, e.g. Mendel-Wang fuzzy reference sets (Wang and Mendel, 1992). The results obtained were of good quality because a more natural inter-dependence between the data set and extracted fuzzy sets was defined. A software package for fuzzy identification and fault detection (based on Mendel-Wang) was developed and is now being evaluated on a boiler node at the Lublin sugar factory in Poland (see Section 5.2.1 for description of the plant and modelling problem). In another study, a *rule-base suitable for the evaluation* of fuzzy-linguistic models (Mamdani and Assilian, 1975) has been used to develop an autonomous mobile vehicle (AMV) (Kovács, Kóczy and Bikfalvi, 1998). The application of these models to the FDI problem has been examined using a real AMV system. Identification using template-based modelling requires two types of information. Firstly, a collection of data points relating to the system inputs and outputs is demanded. Secondly, an expert provides a collection of linguistic values that form a partitioning of the input and output spaces into fuzzy regions. These templates are the system's expert language for describing the environment and the model. The AMV project has progressed towards the use of an additive weighted combination of the rules on the real application (Kovács, Kóczy and Bikfalvi, 1998). The ability of combining identification algorithms, implementation and evaluation of fuzzy identifiers with fuzzy logic models (Sugeno models, neuro-fuzzy structures, evolutionary algorithms of genetic type), appropriate shapes of membership functions make the fuzzy logic approach one of the main approaches to identify models. Garcia *et al.* (1997) used fuzzy implications and reasoning to build fuzzy models for fault diagnosis purposes. The neuro-fuzzy adaptive identification mechanism is applied to real system models.

The fuzzy approach is becoming a powerful alternative to the artificial expert system approach and may gain more practical importance in the future. The non-linear system can be identified using a fuzzy multiple model description of the real system in a parallel and a series model or any combination (series-parallel) (Ballé *et al.*, 1997) and consequently a number of models are identified. The issue that remains a challenge is to obtain not only a number of multiple linear models but also the minimum number of models which describe the non-linear system. This optimisation is difficult because the identification method using fuzzy logic depends on a large number of variables. There are various procedures to try to extract learning rules in combination with other techniques, e.g. using neural networks (Füssel *et al.*, 1997).

4.3. Fuzzy Techniques in FDI

In recent years, the application of fuzzy logic to model-based fault diagnosis approaches has gained increasing attention in both fundamental research and applications. *Symptoms* can be generated using observers based on the estimation of the output from the system. The first methods used fuzzy set theory to express cause-effects relations in expert systems. The key idea of model-based methods is the generation of signals, termed *residuals*. These are usually generated using mathematical methods (based on state observers, parameter estimation or parity equations). The models correspond to the monitored system (Chen and Patton, 1999). Residuals are signals representing inconsistencies between the model and the actual system being monitored, but the deviation between the model and the plant is influenced not only by the presence of the fault but also modelling uncertainty. One solution is for the observer and controller parameters to be tuned via estimation from the real system for fault isolation and threshold adaptation (Schneider and Frank, 1994). The introduction of fuzzy logic can improve the decision-making, and in turn will provide reliable and sufficient FDI, suitable for real industrial applications. However, a difficulty arises in the training of the algorithm in the inference mechanism where knowledge is hidden in large amounts of data and embedded in trained neural networks (Chen *et al.*, 1997b). A fuzzy feed-forward neural network (FNN) is applied to extract rules from an existing data base. Frank *et al.* (Frank 1993; 1994a; 1994b; 1996; Frank and Kuipel, 1993; Frank and Köppen-Seliger, 1997; Schneider and Frank, 1996) use fuzzy logic for residual evaluation. This can be an important way of taking into account modelling uncertainty at decision making rather than during the residual generator design. By applying a fuzzy rule-based approach the fault decision process can be made robust to the uncertainties so that false and missed alarm rates can be minimised. Considering *supervisory control* (Frank and Kuipel, 1993; Linkens and Abbod, 1993) with tasks such as system management, process monitoring, identification, fault detection, diagnosis and adaptive capability reduces at a lower level the models for developing simpler structures for observers and controllers using Takagi-Sugeno models. Several observers had been proposed for FDI in bilinear systems using either unknown input observers (Chen and Patton, 1999), or sliding mode observers (Edwards *et al.*, 1997), or gain-parametrised observers (Bennett *et al.*, 1999). The relationship between the input-output variables can be described by fuzzy qualitative models, fuzzy relational models (using parameters identified from the learning data set) and fuzzy functional models. The Takagi-Sugeno (T-S) fuzzy observers have the advantages of making the error dynamics independent of the parameters of the system (e.g. the speed of the rotor). The (T-S) fuzzy models for non-linear dynamic systems are described by a number of locally-linearised models. In this research, different operating points are self-selected with an optimisation method based on eigenvalue assignment criteria for both a fuzzy observer and a fuzzy controller.

5. Fault Diagnosis Based on Integration of Qualitative and Quantitative Methods

Traditional fault diagnosis and identification methods have limitations, especially when the system is complex and uncertain and the data are ambiguous and not rich in information. Intelligent systems are needed for these complex processes, which can mimic the sensing, generalising, processing, operating and learning abilities of a human operator.

As discussed in Section 4, fuzzy logic offers a linguistic model of the system in which the system can be easily understood by certain rules governing input-output variables. The works (Mamdani and Assilian, 1975; Zadeh, 1965) were pioneering in this field. Bezdek (1981) gave the following definition describing fuzzy partitions: let X a finite set, V_{cn} the set of real $c \times n$ matrices, c an integer, $2 \leq c < n$. Then a fuzzy c-partition space for X is the set

$$M_{fc} = \left\{ U \in V_{cn} | u_{ik} \in [0,1] \, \forall \, i,k; \, \sum_{i=1}^{c} u_{ik} = 1 \, \forall \, k; 0 < \sum_{k=1}^{n} u_{ik} < n \, \forall \, i \right\} \tag{22}$$

Two major classes of knowledge representation in fuzzy modelling are proposed by Takagi and Sugeno (1985) and Mamdani and Assilian (1976). Fuzzy logic has the inherent ability to deal with imprecise or noisy data and neural networks have abilities to learn, generalise and deal with non-linear systems. Here, we propose to integrate symbolic and quantitative knowledge through a neuro-fuzzy system. This will then combine the learning ability of ANNs and the explicit knowledge representation of fuzzy-logic. The application engineer can therefore extract, from the data, a high-level language description of the system. Heuristic knowledge about the plant can also be included.

Some other tools like *evolutionary algorithms, genetic algorithms* or *probabilistic reasoning* can also be combined with the above to enhance the parameter tuning, or to deal with the uncertainty in order to establish the desired intelligent system. Li and Elbestawi (1996) proposed the following four necessary characteristics of an intelligent monitoring system to emulate the human monitoring action:

- indirect sensing,
- signal conditioning,
- parallel processing of information, and
- knowledge learning.

5.1. Combining Neural Networks with Fuzzy Logic

There are several possible methods of combining neural networks with fuzzy logic, the advantages and disadvantages of which depend on the specific application. Some of these techniques are:

1. fuzzification of the inputs or outputs of the neural networks (see Fig. 12),
2. using neural networks in fuzzy models where neurons provide the necessary membership functions and rule base, and
3. fuzzification of the interconnections of conventional neural networks.

Simpson (1992; 1993) proposed a structure for combining neural networks with fuzzy logic, the so-called 'fuzzy min-max neural networks.' The simplest *Fuzzy Neural Networks* use a *fuzzifier* to combine fuzzy logic and a neural network (Caudill and Butler 1992). The fuzzifier is a processor that receives input data patterns and converts them to fuzzy categories which are used as inputs to the neural networks. Fuzzy

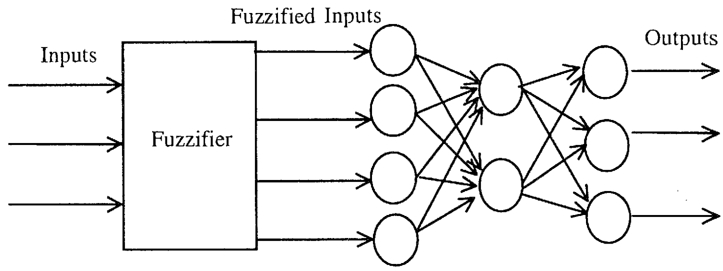

Fig. 12. A simple fuzzy neural network.

logic is most useful when the information available is imprecise or noisy. Fuzzy neural networks can be used for both residual generation and classification.

Many different structures have been established, integrating neural networks with fuzzy logic in the form of fuzzy neural networks with different training algorithms (Jang and Sun, 1995; Lin and Lee, 1991; Wang and Mendal, 1992). Farag *et al.* (1998) proposed a five-layer fuzzy neural network in which the parameter identification of the fuzzy model comprises three phases. In the first phase, initial parameters are found using the Kohonen self-organising feature map algorithm. The second phase consists of finding linguistic rules. In the third phase, a genetic algorithm is used to tune the membership functions, called the multi-resolutional dynamic genetic algorithm (MRD-GA). The neuro-fuzzy network proposed by Farag *et al.* (1998) consists of five layers of fuzzy neurons (Fig. 13). The first layer and five neurons are effectively the input and output neurons, respectively. The second layer together with four neurons act as membership functions and the neurons of the third layer provide a rule base. The training process adjusts the mean value and variance of the bell shaped membership functions represented by Layers 2 and 4, and fuzzy rules by modifying weights between Layers 3 and 4 as described by the diagram of Fig. 13.

Li and Elbestawi (1996) proposed a multiple principal component (MPC) fuzzy neural network for clustering (unsupervised classification) which employs fuzzification of the interconnections of a conventional neural network. This method is used for automated tool condition monitoring in machining and is based on Li and Elbestawi's fuzzy neural networks in which fuzzy membership functions are used for decision making and the interconnection in the network. Figure 14 shows the corresponding neuro-fuzzy architecture. The interconnection between different layers is represented by fuzzy membership functions. The neurons of the hidden layer use fuzzy classification. Note that this structure uses partial interconnection, which results in faster classification.

Neuro-fuzzy methods have been successfully applied to a wide range of applications from industrial processes to financial systems, due to the ease of the design of the rule base, linguistic modelling, application to complex and uncertain systems, inherent non-linear nature, learning abilities, parallel processing and fault tolerance abilities. However, successful implementation of fuzzy neural networks depends heav-

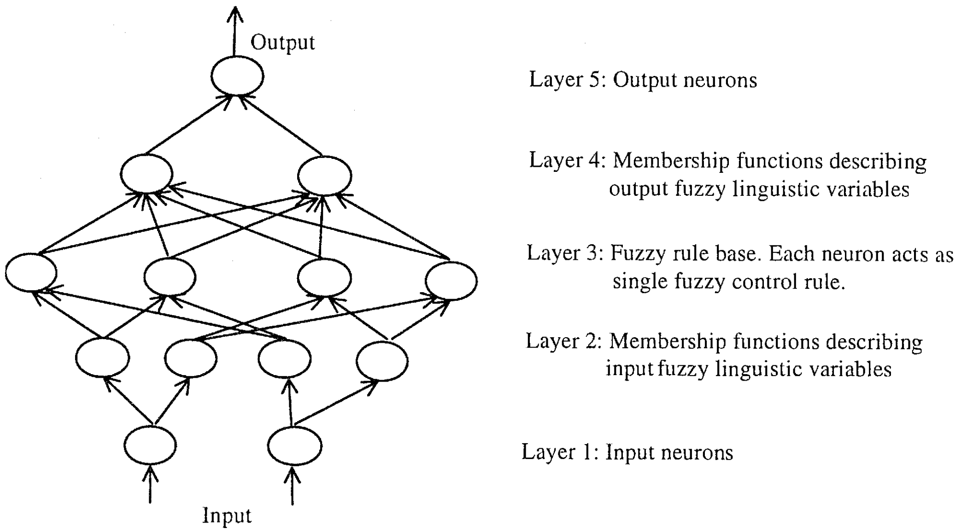

Fig. 13. Fuzzy neural network of (Farag *et al.*, 1998).

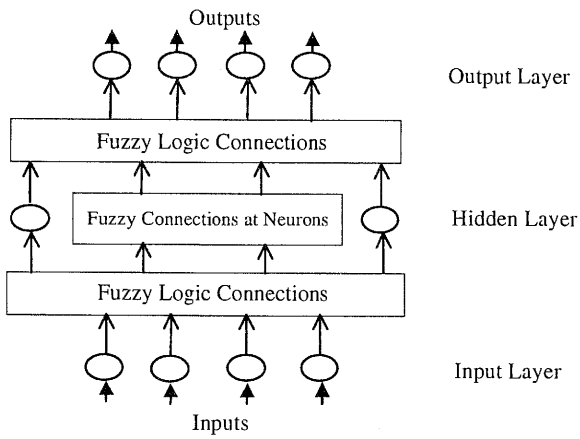

Fig. 14. Fuzzy neural network (Li and Elbestawi, 1996).

ily on the prior knowledge of the system and the training data. Recent research focuses on neural networks, e.g. B-spline neural networks, which can be used to extract the qualitative knowledge of the system.

5.2. B-Spline Neural Networks

Qualitative modelling methods require much less effort, since the requirements for fault diagnosis can be better achieved by symbolic reasoning. The resulting modelling and mathematical operations can be more ambiguous, and the dimension of the model greater than that of the equivalent quantitative approach. For this reason, in order to obtain the advantages of qualitative models, such as effective modelling, flexibility, imprecise numerical information, a qualitative reasoning technique must be combined with other types of knowledge (Kuipers, 1994) or applied in the framework of integration.

By integrating qualitative and quantitative knowledge through a neuro-fuzzy system it is feasible to combine the learning ability of neural networks with the explicit knowledge representation of fuzzy logic. The application engineer can extract from the data a high-level language description of the system and can, if required, include any heuristic knowledge about the plant.

A way of doing so is to use the B-spline neural-network (Brown and Harris, 1994a; Lane *et al.*, 1992). A complete description of the B-spline theory can be found in any general book on spline theory (Ahlberg *et al.*, 1967; Sard and Weintraub, 1971).

B-Spline Residual Generator

As described in Section 3.5, B-spline neural-networks can be used to diagnose faults in non-linear systems and overcome some of the disadvantages of MLP networks. The residual generator is similar to the one proposed by Patton *et al.* (1994). The measured inputs and outputs of the system are processed through an associative memory network, as opposed to an MLP network. The underlying concept is to train the network to recognise the occurrence of a fault and find the optimal function which maps the system inputs and outputs to a residual signal:

$$r(t) = F\big(\vec{u}(t), \vec{y}(t)\big) \tag{23}$$

where $\vec{u}(t) = [u(t),\, u(t-1),\, \dots,\, u(t-m)]^T$ and $\vec{y}(t) = [y(t),\, y(t-1),\, \dots,\, y(t-n)]^T$ are the input and output of the system over a window time, respectively. The input of the network includes past as well as current values of the measurements to capture temporal information.

It is important to note that the aim of the *fault detection observer*, or *residual generator*, is not to estimate the state of the plant but rather to respond promptly to the occurrence of a fault. Hence, the residual generator should output a value of 1 when a fault develops in the system, and 0 otherwise. In such an approach, it may be said that the network used is an alternative to the traditional fault detection observer.

An important feature of a neural-network is that it will learn during a training session made over several training cycles, with training data coming from different operating points. However, before the training is started, the order of the B-spline network needs to be chosen. A network with a second-order basis function and two linear knots is typically chosen. This enables the normalised input space to be divided into three linguistic variables *Small, Medium* and *Large*.

Once the training is performed, a set of optimal weights ω_i is used to derive the corresponding fuzzy description of the residual generator. Moreover, since the basis functions can be interpreted linguistically, a qualitative model of the residual generator can be derived. This provides the operator with an explanation about the cause of the fault which is more understandable to him/her than using crisp neural-networks.

Fault Isolation Using B-Spline Networks

A B-spline network can be used in general FDI schemes which consist in residual generation and decision making (classification). In residual generation, the residual vector $\boldsymbol{r}$ is determined in order to characterise each fault. The residual vector $\boldsymbol{r}$ is then processed to determine the locations and occurrence times of the faults, which is called decision making. Ideally, the models identify all classes of system behaviour.

The B-spline network is used to classify faults in the process. The faults are assumed to be known *a priori*, and their corresponding data available to the designer. The network will then have as many outputs as classes of behaviour. Hence, for a system with two classes of faults, the output of the network will be a three-dimensional vector; this includes the models associated with the two faults as well as that corresponding to the *Healthy* one.

For training, the network is decomposed into a set of $(M+1)$ multi-inputs-single-output (MIMO) sub-models, where M is the number of faulty classes (the set of optimal weighting coefficients for each sub-model can be found). When the network is used to classify a test point $(\vec{u}(t), \vec{y}(t))$, the network's output *Flag* is a real vector of dimension $(M+1)$.

It can be seen from Fig. 16 that each component of that vector, $Flag_I$ $(I = 1, 2, \ldots, M)$ is identified with a class of behaviour, which can be either $Fault\ I$, or the nominal model. When the system is operating in its nominal condition, all the network outputs are zero except the last one. However, when a specific fault develops in the system, the corresponding output will deviate from zero, whereas the output $Flag_{I+1}$ becomes zero, confirming that the system is no longer *Healthy*.

5.2.1. B-Spline Neural Networks Applied to a Sugar Factory Plant

An interesting example of the use of the B-spline network has been the modelling and diagnosis of some stages of the Lublin sugar factory in Poland. The factory illustrated in Fig. 17 produces 50,000 thousand tonnes of sugar annualy. It consists of a large number of evaporation plants, boiler houses, heaters and valves.

For the production of sugar, beets are collected from the planters. Raw syrup is obtained from thin sliced beets using *extractors*. After cleaning, decalcifying and processing the syrup, it contains 14 % of sugar condensed to 70 % solution using the evaporation station. The waste steam from the steam turbine is used as a heat source for the complete process. A mixture of granulated sugar and syrup is obtained after crystallisation process in syrup boilers. The main by-products of this process are the beet-pulp, used for feeding cattle and the remaining syrup used for manufacturing alcohol and organic acids. An automatic control system OSA-2 is used to operate and

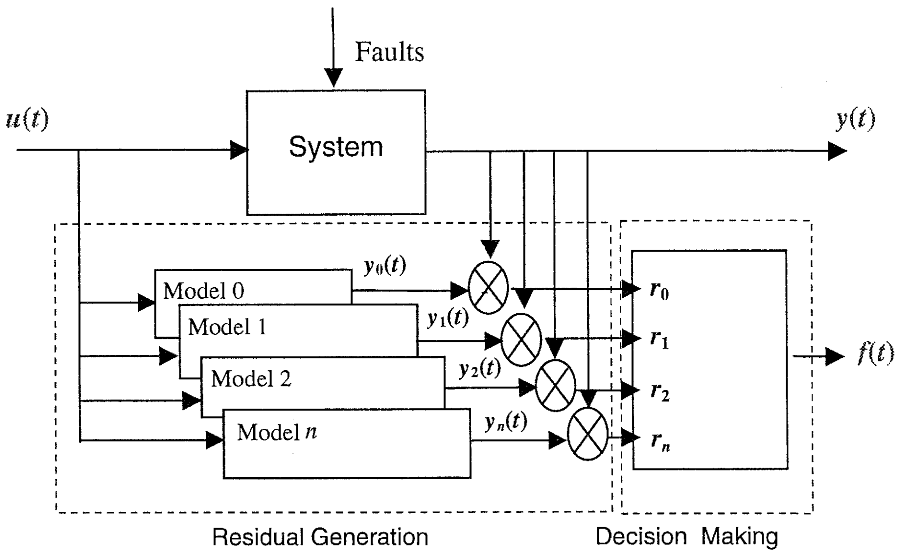

Fig. 15. General multiple-model FDI scheme.

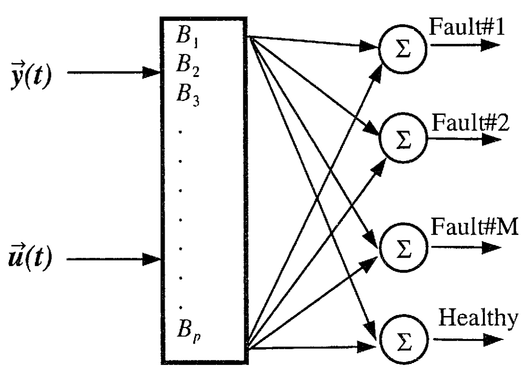

Fig. 16. Network-architecture used for fault-isolation.

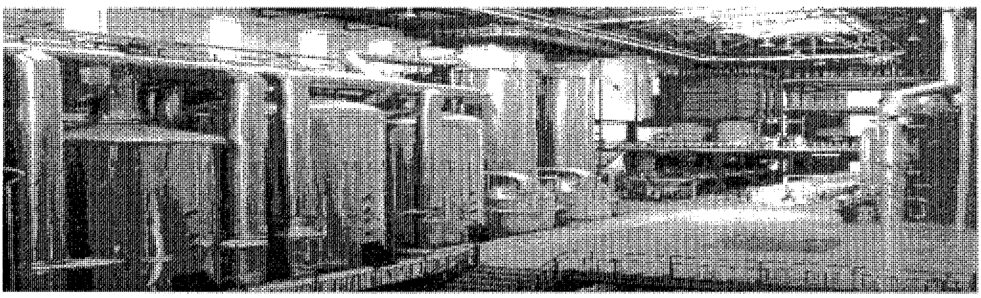

Fig. 17. Sugar factory plant in Lublin.

monitor the system. The whole process is optimised to consume minimal heat. Steam is generated by the boiler house, and is composed of four technological complexes. Each complex consists of a coaling grate furnace having two grates of OSR 32 type, barrel boiler and initial steam super-heater. The thickness of the coal coat is controlled in an open-loop manner, via a boiler operator.

The steam pressure in the barrel boiler determines the feed of each grate and the air inflate through coal in each grate (Wasiewicz, 1999). Real data from the plant were taken and a B-spline network was applied to identify various sub-parts including a valve controlling the flow of the juice into the evaporisation plant.

Figures 18 and 19 show the result of applying the B-spline network to the training and testing data for the identification of the valve mentioned above. Note that the result when using the training data is much better when compared with the result using the testing data because it contains different levels of noise and disturbances and the training is never perfect. The initial large error in the testing result is due to a wrong initial condition on the predicted output. At the commencement of the algorithm, the B-spline uses a feedback from its own output value, which does not match with the output of the actual system. This error reduces quickly as the B-spline network output converges to that of the system. From Fig. 19 it can also be observed that the system does not perform well if it operates outside the input-output space of training data.

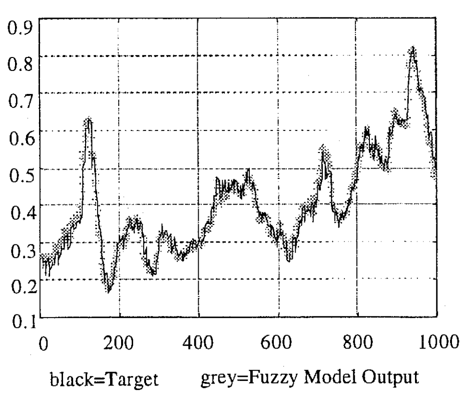

Fig. 18. Result of training data.

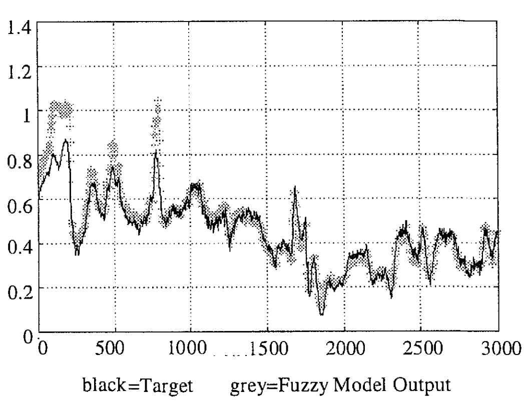

Fig. 19. Result of testing data.

5.2.2. Extracting Fuzzy Rules from the B-Spline Network

As regards the structural equivalence of B-spline and fuzzy logic, it can be seen that both have input membership functions and *weighted sums of product* of these functions. Consider the following fuzzy rule:

$$R_{ij}: \quad \text{IF } (x \text{ is } A_i), \quad \text{THEN } (r \text{ is } B_j) \qquad (c_{ij}) \tag{24}$$

where A_i and B_j denote fuzzy sets in the input and output partition space, respectively, and c_{ij} is the level of confidence in rule R_{ij} being true (Benkhedda and Patton, 1997).

The output of a continuous fuzzy rule can be described as

$$r(x) = \sum_{i=1}^{p} \mu_{A_i}(x)\omega_i \tag{25}$$

where $\omega_i = \sum_{j=1}^{q} c_{i,j} y_j^c$. The confidence level of each rule can be obtained from optimal B-spline network weights (Brown and Harris, 1994a) using the equation

$$c_{ij} = \mu_{B_j}(\omega_i) \tag{26}$$

Figure 20 shows the linguistic fuzzy rules extracted from the B-spline network and Fig. 21 shows the confidence level of each rule connecting the i-th input function with the j-th output function. In fact, the qualitative knowledge about the system in the form of fuzzy rules is not useful in this form to offer an insight of how the system works. A graph showing the relation between the output and input of the system at different operating points is much useful to understand the operation of the model. For example, Figs. 22 and 23 show the dynamical relation between the output flow and the two input variables.

Figure 24 shows the relation between the number of inputs in the B-spline network and the time of learning. This exponential curve means that for a given computational time, the number of inputs to be used in the B-spline network is limited. In a manner similar to other neuro-fuzzy structures, the B-spline network suffers from the 'curse of dimensionality.' For large numbers of inputs, the number of rules governing the behaviour of the system also becomes very large and it is very difficult to visualise the operation of such a system. There is no guarantee of predicting correctly in the case of data applied outside the input-output space for training (see Fig. 19).

6. Non-Linear FDI via Fuzzy Observers

6.1. Takagi-Sugeno Fuzzy Models: A Connection

It is possible to establish the equivalence of a generalised class of Gaussian RBF networks and the Takagi-Sugeno model of fuzzy inference. A standard Gaussian RBF and a restricted form of the T-S model of fuzzy inference are functionally equivalent (Jang and Sun, 1993). The standard RBF network is functionally equivalent to the T-S fuzzy inference model under the following conditions:

1. There are some conditions required to make the RBF and the fuzzy inference system structurally equivalent (Hunt *et al.*, 1996), e.g. the number of RBF units must be equal to the number of if-then rules, the T-norm being the operator used to compute each rule's firing is *multiplication* and the method to derive their overal outputs for the RBF network should be the same method for the T-S model.

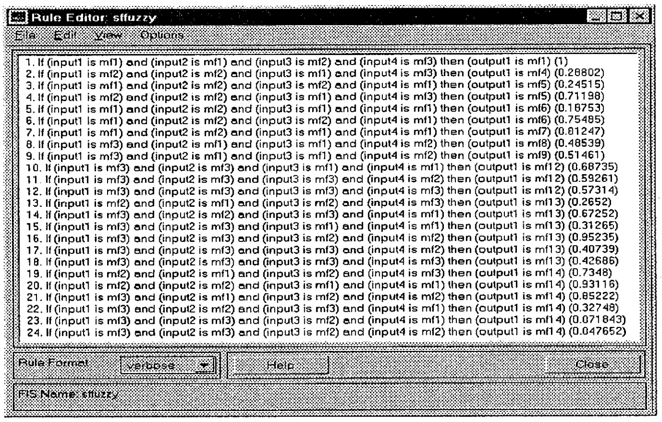

Fig. 20. Fuzzy rules governing the system.

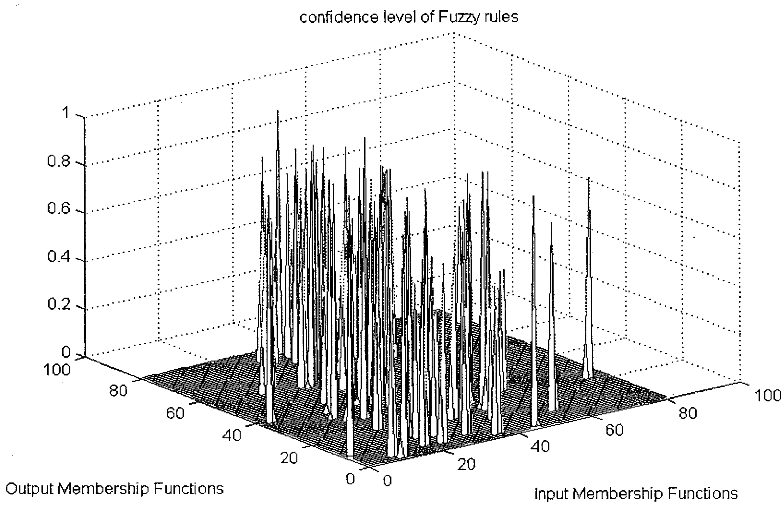

Fig. 21. Confidence level of the extracted fuzzy rules for Valve 1 in matrix form.

2. In order to restrict the network to the class of T-S structure some conditions must be satisfied: The output of each fuzzy if-then rule is a constant and the membership functions chosen have to be Gaussian with the same width.

For T-S models, stability conditions and a pole assignment in LMI regions are derived in Lopez-Toribio *et al.* (1999).

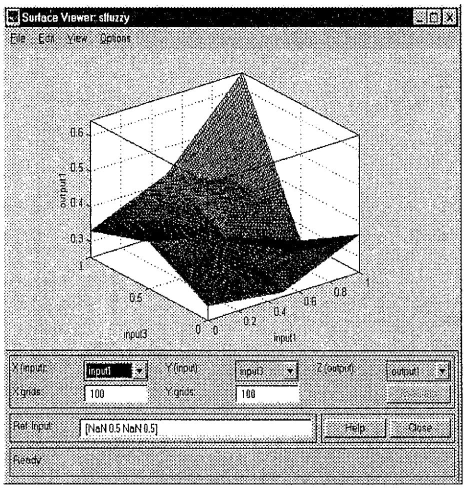

Fig. 22. Relation between the pressure (I/P1), previous value of O/P (Input 3) and output.

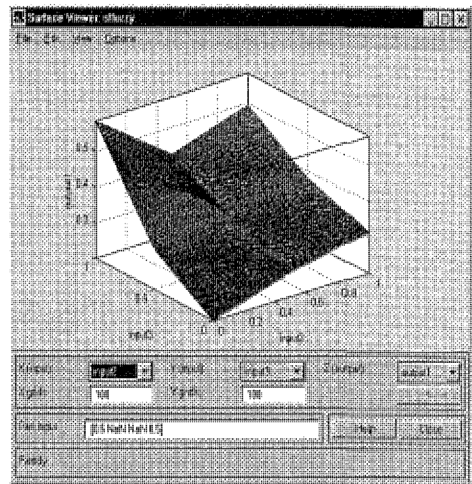

Fig. 23. Relation between the control (I/P1), previous value of O/P and output.

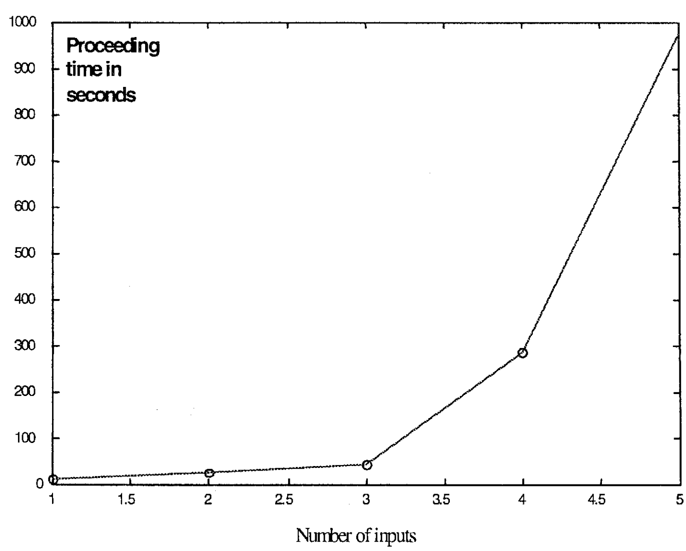

Fig. 24. Computational time increases exponentially with inputs.

6.2. Adaptive T-S Fuzzy Observers for Non-Linear Systems

The classical model-based fault diagnosis is altered in order to incorporate a new adaptive scheme. The model-based fault diagnosis (Fig. 25) consists of a parameter estimation block to estimate unknown measurements. Also, the new scheme incorporates an adaptive law based on the plant parametric model for tuning the observer.

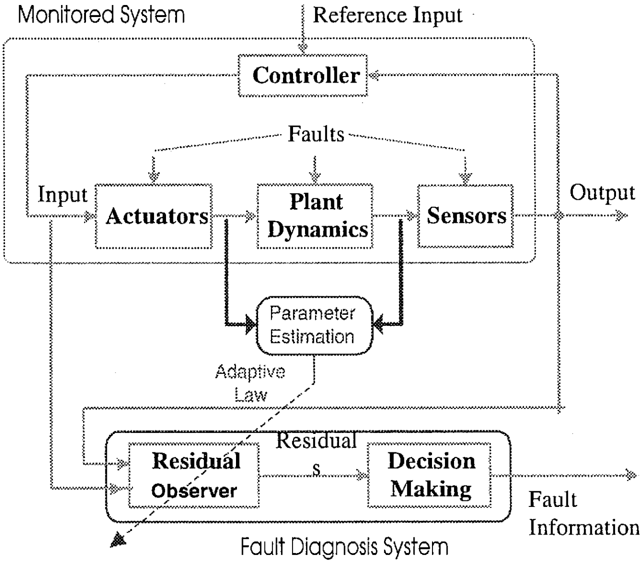

Fig. 25. Schematic diagram of an adaptive FDI system.

For a non-linear dynamic system described by the T-S fuzzy model, a fuzzy observer can be designed to estimate the system state vector. For fuzzy observer design, it is assumed that the fuzzy system model is locally observable, i.e. all (A_x, C_x) pairs $(x = 1, \dots, r)$ are observable. Using the idea of *parallel distributed compensation* (the use of parallel dissimilar feeback paths, each one corresponding to a different model (Tanaka *et al.*, 1996; Wang *et al.*, 1995)), for a non-linear dynamic system represented by the T-S fuzzy model a linear time-invariant observer can be associated with each rule of the fuzzy model:

IF $(\omega(t)$ is $M_x)$ THEN

$$\begin{cases} \dot{\hat{x}}(t) = A_x \hat{x}(t) + B_x u(t) + L_x\big(y(t) - \hat{y}(t)\big) \\ \hat{y}(t) = C_x \hat{x}(t) \end{cases} \tag{27}$$

The overall observer dynamics will then be a weighted sum of individual linear observers:

$$\begin{cases} \dot{\hat{x}} = \sum_{x=1}^{r} \mu_x(w)\big[A_x x + B_x u + L_x(y - \hat{y})\big] \\ \hat{y} = \sum_{x=1}^{r} \mu_x(w) C_x \hat{x} \end{cases} \tag{28}$$

where $\mu_x(w(t))$ is the grade of membership of the premise variable, $w(t)$, or the tensor product of grade of memberships, if $w(t)$ is a vector. The membership grade

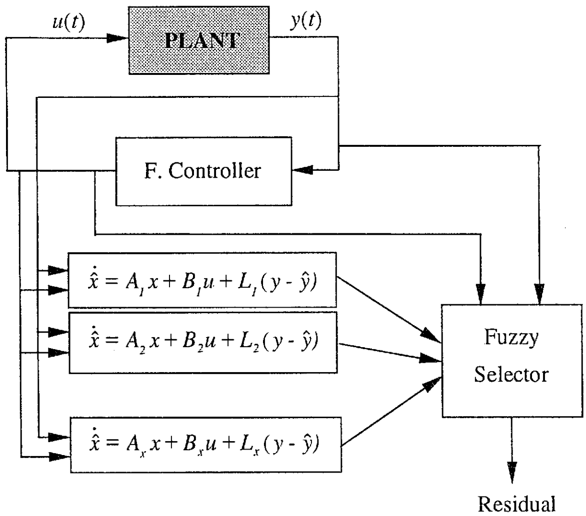

Fig. 26. Schematic diagram of the fuzzy observer.

function $\mu_x(w(t))$ satisfies the following constraints:

$$\sum_{x=1}^{r} \mu_x\big(w(t)\big) = 1, \quad 0 \leq \mu_x\big(w(t)\big) \leq 1 \quad \forall x = 1, 2, \ldots, r$$

The schematic diagram of such an observer is shown in Fig. 26 where it can be seen that a fuzzy inference engine is used to 'select' the appropriate output from those generated by the r parallel observers. The transition between one model to another depends on the operating regime defined by ω. The estimation error dynamics can then be shown to be given by the following differential equation:

$$\dot{e}(t) = \sum_{x=1}^{r} \mu_x (A_x - L_x X_c) e(t) \tag{29}$$

If the above error dynamic equation is stable, the state estimation will asymptotically converge to the real state. An observer with converging state estimation can also be referred to as a stable observer. It can be proved that the stability of the above error dynamic equation can be verified by the following theorem.

Theorem 1. *The fuzzy observer given by (27) has all its eigenvalues in D if and only if there exists a symmetric matrix P such that*

1) $$M_D^x(\bar{A}_x, P) = [\phi_{ij} P + \theta_{ij} \bar{A}_x P + \theta_{ij} P \bar{A}_x^T]_{1<i,\, j<k} < 0$$

$$\bar{A}_x = (A_x - L_x C_x) \qquad \textit{for} \quad x = 1, \ldots, r \tag{30}$$

and

2) $$M_D^x(\bar{A}_y, P) = [\phi_{ij}P + \theta_{ij}\bar{A}_yP + \theta_{ij}P\bar{A}_y^T]_{1<i,\, j<k} < 0$$

$$\bar{A}_y = \left(\frac{A_y - L_yC_x + A_x - L_xC_y}{2}\right) < 0 \qquad \textit{for} \quad x < y \le r \tag{31}$$

Using the results presented in this section, it is possible to derive the conditions for a stable observer. More than that, it is also possible to assign the eigenvalues of the observer to a region in the s-plane.

The fuzzy observer given by eqn. (27) can be simplified if there is no uncertainty and non-linearity involved in the system output equation, i.e. $C_1 = C_2 = \cdots = C_r = C$ and $Y(t) = Cx(t)$. This is a very common situation in practice because the system dynamics is not involved in the output equation. The simplified fuzzy observer is given by

$$\begin{cases} \dot{\hat{x}} = \sum_{x=1}^{r} \mu_x(w)\left[A_x x + B_x u + L_x(y - \hat{y})\right] \\ \hat{y} = C\hat{x} \end{cases} \tag{32}$$

Corollary 1. *The simplified fuzzy observer of eqn. (32) has all its eigenvalues in D if and only if there exists a symmetric matrix P such that*

$$M_D^x(A_x - L_xC_x, P) = \left[\lambda_{ij}P + \mu_{ij}(A_x - L_xC_xP) + \mu_{ji}(PA_x - L_xC_x)^T\right]_{1<i,j<k} < 0 \qquad \textit{for} \quad x = 1, \ldots, r \tag{33}$$

For both the full and the simplified observer, a minimum number of models must be found along with the required operating points $w_{r1}, w_{r2}, \ldots, w_{rn}$ with suitable widths (Fig. 27), so that a convex optimisation technique involving LMI can be used to find the optimum set of models (matrices A_x, B_x, C_x, D_x) used in the fuzzy-observer representation, estimating the parameters of the system for each operating point. The optimisation method consists in solving the inequalities for each step until the solution is found. In this stage, the LMI solver goes back and solves for half a step until it is possible to solve all the inequalities for all the linear models.

The completed adaptive scheme is illustrated in Fig. 28. The supervision block is required not only because of the decision making but also because the design of the fuzzy observer must be achieved according to two subtasks: the restriction of the eigenvalues must be assigned to a specific region to ensure system stability in the presence of uncertainties. A fast state estimation response is also achieved for a good response to faults. Once the system parameters are found, the LMI solver calculates the matrices for each linear model. The model parameters are the matrices used for the design of the adaptive fuzzy observer.

Qualitative fault diagnosis. Fault diagnosis of dynamic systems can also be based upon declarative knowledge of the system, which is available in qualitative rather than quantitative form (Howell, 1994; Leitch 1993; Shen and Leitch, 1993; Zhuang

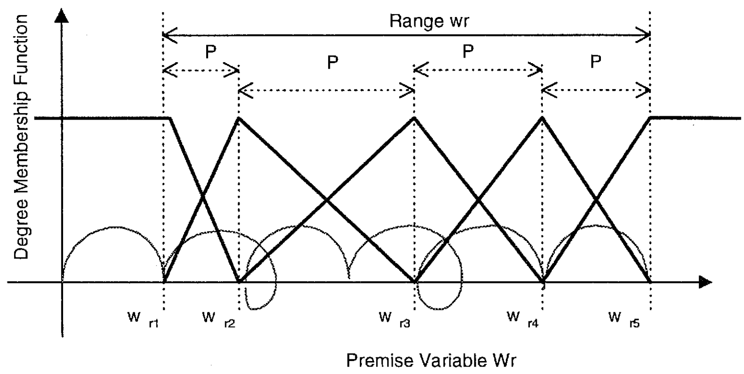

Fig. 27. Fuzzy membership functions.

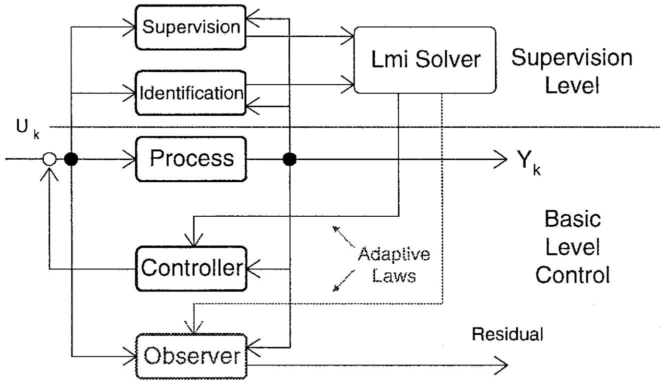

Fig. 28. Fuzzy adaptive observer scheme plus adaptive controller.

and Frank 1997). The qualitative approach is based upon the concept of a qualitative model described by means of fuzzy rules which unlike the quantitative counterpart, only require declarative (heuristic) information. A fuzzy qualitative observer (Zhuang and Frank, 1997) can be designed (see Fig. 29) making use of the fuzzy qualitative simulation in order to produce a residual generator.

The qualitative model of the process can be seen as an observer. Since the model used to obtain the observations of the process is qualitative, the states (behaviours) will be qualitative. The qualitative states are obtained via simulation or via observation.

In order to reduce the ambiguity resulting from qualitative simulation, qualitative observers are used to generate the predictions of the possible qualitative states. In FDI, the qualitative observer can be used as a substitute when some quantitative information of the process is not available. A Fuzzy Qualitative Simulator developed

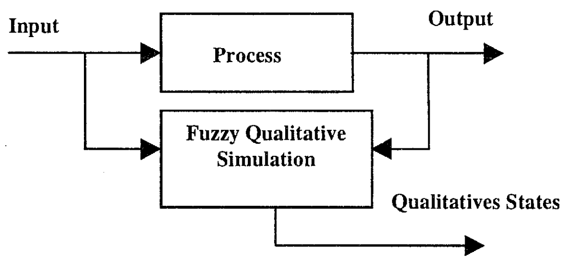

Fig. 29. Fuzzy adaptive observer scheme plus adaptive controller.

by Heriot-Watt University, Edinburgh, called *FuSim* (fuzzy interval-based simulator) was proposed by Shen and Leitch (1993). This simulator presents a methodology to integrate knowledge of the common sense and the qualitative simulation of physical systems by means of fuzzy sets. The use of an amount of fuzzy space facilitates and allows for a detailed description of the relations between two or more variables. This method produces a reduction of the set of spurious behaviours by means of temporary filters, although these behaviours continue to exist for complex systems. Like Q2, FuSim uses the Taylor-Lagrange formula for temporary calculations, producing identical problems.

6.3. Robustness of Fuzzy Systems in FDI

Model-based fault diagnosis is based upon the use of mathematical models of the supervised system. Modelling errors and other uncertainties (e.g. unknown disturbances) are inevitable when model-based methods are applied to complex systems. Hence, there is a need to develop robust fault diagnosis algorithms which take into account some or all of the uncertainties. The robustness of a fault diagnosis system means that it must be only sensitive to faults, even in the presence of model-reality differences (e.g. parameter variations and disturbances). The uncertainty may be difficult to account for and the design of a fault diagnosis system which is highly sensitive to faults, whilst insensitive to uncertainty and unmodelled disturbances, remains a real challenge (Chen and Patton, 1999; Frank and Ding, 1997). So far the robustness problem has been addressed only by an empirical analysis of the results of selected simulations. A general outcome is that fuzzy systems are very robust against parameter changes. This observation can be explained by the fact that fuzzy systems contain strong non-linearities. A systematic theoretical investigation of the robustness of fuzzy systems should be performed. It is difficult to establish and analyse the sensitivity of the control-loop as a whole. The study can be realised via two main approaches. The first approach consists in transforming the description of the plant into a fuzzy model. The other approach is concerned with transforming the controller into a mathematical description. This method has the advantage of applying existing methods to the analysis of the system. Klotzek *et al.* (1998) carried out the approach of applying sensitivity theory to a fuzzy PI control loop. The entire method used by

Klotzek *et al.* (1998) for the application of sensitivity theory to fuzzy logic is divided into three main tasks (representation of fuzzy systems mathematically, application of the theory to the mathematical representation, and determination of the sensitivity with respect to parameter changes of the fuzzy model).

6.4. Evolutionary Algorithms of Genetic Type

Neural networks can be trained to replicate dynamic system behaviour during normal and abnormal operation. A neural network behaves as an *implicit* model of the process (because a mathematical model of the process is not actually required). In order to assure a good accuracy of these models, the neural network structures must be optimised. The research has shown that evolutionary techniques cope efficiently with this optimisation problem. They can be used in order to implement semi-automatic procedures, dedicated to the selection of convenient neural network topologies and parameters. Many papers have focused on the development of evolutionary-based algorithms for two types of neural network structures. The first one is applied to a feed-forward network structure and the other is applied to the dynamic multi-layer perceptron. 'Near-optimal' neural network topologies can be obtained by minimising their complexity order and the corresponding output-squared-error can be computed for the whole training data set. The proposed procedures are used for an appropriate construction of neural observer schemes, in order to perform a robust diagnosis (detection and fault isolation) of the process faults. The user must set some parameters of a genetic algorithm (GA), but this seems to be easier than manually selecting the neural network topology. An advanced study of network optimisation using evolutionary programming and GA approaches has recently been reported (Obuchowicz and Korbicz, 1998; Obuchowicz and Patan, 1997).

Genetic algorithms have also been successfully used to optimise the design of model-based observers for residual generation (Patton *et al.*, 1997). This study used a multi-objective approach with objectives corresponding to various sensitivity and robustness design issues to achieve a good residual response to faults and minimise the effects of disturbance and noise acting at different frequencies. This approach can be contrasted with the use of GAs for neural-network optimisation.

7. Conclusions

AI approaches to fault diagnosis can be very effective in enhancing the powerful detection and isolation capabilities of quantitative model-based methods. This paper has focused on a discussion of the integration of qualitative and quantitative strategies to minimise the probability of false-alarms and missed-alarms in fault decision-making, whilst improving the level of heuristic information available for the human operator. Residual-based methods for FDI most often use state observers, Kalman filters, but there is a growing tendency to substitute the use of the model-based observer/estimator by a neural network which needs no explicit model for construction and training. The neural network is, on the other hand, an implicit or 'black box' model which does not give a simple insight into the sort of system behaviour which is important for diagnosis.

The main emphasis of the paper has been the simple point that by combining together a fuzzy rule-based strategy with a neural network some powerful diagnostic results can be obtained. This is especially true when considering the diagnosis of complex systems which are hard to model (e.g. the sugar factory evaporisation plant). The advantage of using fuzzy logic is that it supports, in a natural way, the direct integration of the human operator into the fault detection and supervision process using rules which are easy to understand. Fuzzy-logic methods are rapidly becoming a powerful alternative to the use of artificial expert systems.

The combination of neural networks and fuzzy logic for the purpose of fault diagnosis is nothing but the integration of quantitative and qualitative methods. The so-called fuzzy neural network (FNN) takes the advantages of neural networks in adaptation of knowledge learning, distributed parallel processing of data, associative memory and distributed storage of diagnosis rules, to overcome the difficulties of expert systems in the knowledge acquisition bottleneck and the knowledge inference matching conflict. The FNN also takes the advantage of fuzzy logic in knowledge fuzzy reasoning to overcome (at least in part) the black box limitation of the neural network.

Several approaches to FNNs have been outlined and the paper has provided a limited survey of some world-wide studies. Of key importance in the literature is the use of the Mendel-Wang and B-spline networks, both of which provide powerful FNN structures for diagnostic reasoning. Some useful results of the application of B-spline networks for modelling and diagnosis of the evaporisation plant of a sugar factory have also been outlined, as a real system example. A part of this research has been funded by the EC INCO-Copernicus project IQ^{2}FD (the Integration of Qualitative and Quantitative Methods for Fault Diagnosis) in which eleven partner groups have developed, compared and contrasted various methods of integrating qualitative and quantitative methods for FDI, with a focus on the use of data from the sugar factory.

Finally, the Takagi-Sugeno approach to multiple-model observer design for FDI has been outlined. This incorporates fuzzy rules, based on easily understood premise variables, with state space models dependent on the operation point. This powerful combination of fuzzy logic and quantitative modelling provides a robust solution for FDI, minimising false-alarms and missed detection of faults, in the presence of disturbance and changes in plant operation.

Acknowledgements

This work has been funded in part by the EC INCO-Copernicus project *Integration of Quantitative and Qualitative Fault Diagnosis Methods within the Framework of Industrial Application* (IQ^{2}FD) through which the Lublin sugar factory study has been made possible. The rail traction system study has been conducted in collaboration with ABB-Alstom using a three-phase induction motor test rig. Faisel Uppal acknowledges funding support from the UK CVCP ORS (Overseas Research Scholarship) fund and the University of Hull Open Scholarship.

References

Ahlberg J.H., Nilson E.N. and Walsh J.L. (1967): *The Theory of Splines and Their Applications.* — New York: Academic Press.

Ayoubi M. (1995): *Neuro-fuzzy structure for rule generation and application in the fault diagnosis of technical processes.* — Proc. American Control Conference, Washington, pp.2757–2761.

Babuška R. (1997): *Fuzzy Modelling and Identification.* — Ph.D. Th., Delft University of Technology, Delft, the Netherlands.

Ballé P., Nells O.C. and Füssel D. (1997): *Fault detection for non-linear process based on local linear fuzzy models in parallel and series-parallel model.* — Proc. IFAC Symp. Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'97, University of Hull, UK, Pergamon Press, pp.1137–1142.

Bellman R.E. and Zadeh L.A. (1970): *Decision-making in a fuzzy environment.* — Management Sci., Vol.17, No.4, pp.141–164.

Benkhedda H. and Patton R.J. (1997): *Information fusion in fault diagnosis based on B-spline networks.* — Proc. IFAC Symp. Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'97, University of Hull, UK, Pergamon Press, pp.681–687.

Bennett S.M., Patton R.J. and Daley S. (1999): *Sensor fault-tolerant control of a rail traction drive.* — Contr. Eng. Practice, February 1999, Vol.7, pp.217–225.

Bezdek J.C. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms.* — New York: Plenum Press.

Billings S. and Leontaritis I. (1985): *An input-output parametric model for non-linear systems Part I: Deterministic non-linear systems.* — Int. J. Contr., Vol.41, No.2, pp.303–328.

Boucherma M. (1995): *Turbo-Generator Fault-Detection and Diagnosis Based on Artificial Neural Networks.* — Ph.D. Th., University of Sheffield, UK (No.45–11736).

Braake H., Babuška R. and van Can E. (1994): *Fuzzy and neural models in predictive control.* — J. A., Vol.3, No.35, pp.44–51.

Brown M. and Harris C.J. (1994a): *Neuro-Fuzzy Adaptive Modelling and Control.* — New York: Prentice Hall.

Brown M. and Harris C.J. (1994b): *The modelling abilities of the binary CMAC.* — IEEE Int. Conf. Neural Networks, Orlando, USA, pp.1335–1339.

Brown M. and Harris C.J. (1994c): *Neural networks for modelling and control.* — Advances in Intell. Contr., Taylor and Francis, London, pp.18–55.

Calado J.M.F. and Sa da Costa J.M.G. (1999): *A fault detection and diagnosis methodology for chemical processes.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.O, pp.539–544.

Carley M.P. (1997): *Dynamic Neural Networks for Time Series Modelling with Application to Power System Fault Detection.* — Ph.D. Th., University of Virginia.

Catfolis T.Y. (1996): *Artificial Neural Networks in the Process Industry: Concepts and Applications (Rtrl, Control).* — Ph.D. Th., Katholieke Universiteit Leuven (Belgium).

Caudill M. and Butler C. (1992): *Understanding Neural Networks: Computer Explorations.* — Massachusetts Institute of Technology.

Chen J. and Patton R.J. (1999): *Robust Model Based Fault Diagnosis For Dynamic Systems.* — Dordrecht: Kluwer.

Chen J., Patton R.J. and Liu G.P. (1997a): *Robust fault detection of dynamic systems via genetic algorithms.* — Proc. Instn. Mech. Engrs., Vol.211, Part I, pp.357–364.

Chen J., Patton R.J. and Zhang H. (1996): *Design of unknown input observers and robust fault detection filters.* — Int. J. Contr., Vol.63, No.1, pp.85–105.

Chen B.H., Shang Z.G., Yang S.H., Wang X.Z. and McGreavy C. (1997b): *Operational support system for on-line faults monitoring in chemical manufacturing.* — Proc. IFAC Symp. *Fault Detection Supervision and Safety for Technical Processes, SAFEPROCESS'97*, Hull, UK, Pergamon Press, pp.932–937.

Chiang C.Y. (1996): *Neural Network And Fuzzy Logic Approach On Aircraft Failure Detection, Isolation And Reconfiguration Of Controls.* — Ph.D. Th., University Of Southern California.

Crowther W.J., Edge K.A., Burrows C.R., Atkinson R.M. and Woollons D.J. (1998): *Fault diagnosis of a hydraulic actuator circuit using neural networks an output vector space classification approach.* — J. Syst. Contr. Eng., Vol.212, No.1, pp.57–68.

Dalmi I., Kocvacs L., Lorant I. and Terstyansky G. (1999): *Application of supervised and unsupervised learning methods to fault diagnosis.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.P, pp.91–96.

De Kleer J. and Williams B.C. (1987): *Diagnosis multiple faults.* — Artif. Intell., Vol.32, pp.97–130.

Dexter A.L. (1995): *Fuzzy model-based fault diagnosis.* — IEE Proc. Contr. Th. Appl., Vol.142, No.6, pp.545–550.

Doherty, Gomm B., Eardley W. (1994): *Design Issues in Applying Neural Networks to Model Highly Non-Linear Processes.* — Proc. IEE Int. Conf. *Control 94*, Short Run Press Ltd, Exeter, UK, IEE Conference Publication No.389, pp.1478–1483.

Dong D. and McAvoy T.J. (1996): *Non-linear Principal Component Analysis- Based on Principal Curves and Neural Networks.* — Comp. Chem. Eng., Vol.20, pp.65–78.

Edelmeyer A., Bokor J. and Keviczky L. (1994): *An H_∞ filtering approach to robust detection of failures in dynamical systems.* — Proc. 33rd IEEE Conf. *Dec. Contr.*, Orlando, FL, pp.3037–3039.

Edelmeyer A., Bokor J. and Keviczky L. (1997): *Improving sensitivity of H detection filters in linear systems.* — Proc. IFAC Symp. *Systems Identyfication, SYSID'97*, Kytakiushu, Japan, pp.1195–1200.

Edwards C., Spurgeon S.K., Patton R.J. and Klotzek P. (1997): *Sliding mode observers for fault detection.* — Proc. IFAC Symp. *Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'97*, University of Hull, UK, Pergamon Press, pp.507–512.

Farag W.A., Quintana V.H. and Lambert-Torres G. (1998): *A genetic-based neuro-fuzzy approach for modelling and control of dynamical systems.* — IEEE Trans. Neural Networks, Vol.9, No.5., pp.756–767.

Forbus K.D. (1984): *Qualitative Process Theory.* — Artif. Intell., Vol.24, pp.85–168.

Ford N. (1991): *Expert Systems and Artificial Intelligence: An Information Manager's Guide.* — London: Library Association.

Franckin R.E., Mariela C.L. and Danny A.M. (1999): *Fault detection scheme using neural networks with fuzzy preprocessing.* — Proc. IFAC 14th World Congress, Vol.K, pp.379–382.

Frank P.M. (1993): *Advances in observer based fault diagnosis.* — Proc. Int. Conf. *Fault Diagnosis: TOOLDIAG'93*, Toulouse, France, pp.817–836.

Frank P.M. (1994a): *Application of fuzzy logic process supervision and fault diagnosis.* — Pre-prints of the IFAC Symp. *Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'94*, Espo, Finland, pp.531–538.

Frank P.M. (1994b): *On-line fault detection in uncertain non-linear systems using diagnostic observers - a survey.* — Int. J. Syst. Sci., Vol.25, No.12, pp.2129–2154.

Frank P.M. (1996): *Analytical and qualitative model-based fault diagnosis — A survey and some new results.* — Europ. J. Contr. Vol.2, No.1, pp.6–28.

Frank P.M. and Ding X. (1997): *Survey of robust residual generation and evaluation methods in observer-based fault detection systems.* — J. Proc. Contr., Vol.7, No.6, pp.403–424.

Frank P.M. and Köppen-Seliger B. (1997): *Fuzzy logic and neural network applications to fault diagnosis.* — Int. J. Appr. Reas., Vol.16, No.1, pp.67–88.

Frank P.M. and Kuipel N. (1993): *Fuzzy supervision and application to lean production.* — Int. J. Syst. Sci., Vol.24, No.10, pp.1935–1944.

Füssel D., Ballé P. and Isermann R. (1997): *Closed-loop fault diagnosis based on a nonlinear process model and automatic fuzzy rule generation.* — Proc. IFAC Symp. *Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'97*, University of Hull, UK, Pergamon Press, pp.359–364.

Garcia J., Izquierdo V., Miguel L.J. and Peran J. (1997): *Fuzzy identification of systems and its application.* — Proc. IFAC Symp. *SAFEPROCESS'97*, University of Hull, UK, Pergamon Press.

Gertler J. (1998): *Fault Detection and Diagnosis in Engineering Systems.* — New York: Marcel Dekker.

Hagan M.T., Demuth H. and Beale M. (1996): *Neural Network Design.* — Boston: PWS Publishing Company.

Haykin S. (1994): *Neural Networks. A Comprehensive Foundation.* — New York: McMillan College Publishing Co.

Hennerberger D., Patton R.J., Chen J., Wolff A. and Köppen-Seliger B. (1993): *A neural-network with an adaptive grid for fault detection.* — IEE Colloquium: Advances in Neural-Networks for Control and Systems, University of Reading, UK.

Hoskins J.C. and Himmelblau D.M. (1988): *Artificial neural network models of knowledge representation in chemical engineering.* — Comp. Chem. Eng., Vol.12, No.9, pp.881–890.

Howell J. (1994): *Model-based fault detection in information poor plants.* — Automatica, Vol.10, No.6, pp.929–943.

Hu B.G., Mann G.K.I. and Gosine R.G. (1999): *How to evaluate fuzzy PID controllers without using process information.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.K, pp.177–182.

Hunt K.J., Haas R., Murray-Smith R. (1996): *Extending the functional equivalence of radial basic function networks and fuzzy inference systems.* — IEEE Trans. Neural Networks, Vol.7. No.3, pp.776–781.

Hunt K.J., Sbarbaro D., Zbikowski R. and Gawthrop P. (1992): *Neural networks for control systems: A survey.* — Automatica, Vol.28, No.6, pp.1083–1112.

Ilott P.W. and Griffiths A.J. (1997): *Fault diagnosis of pumping machinery using artificial neural networks.* — J. Process Mech. Eng., Vol.211, No.3, pp.185–194.

Isermann R. (1994a): *Fault diagnosis of machines via parameter estimation and knowledge processing — A tutorial paper.* — Automatica, Vol.29, No.4, pp.815–835.

Isermann R. (1994b): *Process fault detection and diagnosis methods.* — Proc. IFAC Symp. *SAFEPROCESS'94*, Helsinki, Finland, Vol.2, pp.597–612.

Isermann R. and Ulieru M. (1993): *Integrated fault detection and diagnosis.* — Proc. IEEE Syst., Man Cybern., le Touquet, France, pp.743–748.

Isoc D. (1998): *On a new approach to build the membership functions for fuzzy models.* — Proc. CONTI '98, Timisoara, Romania.

James Li C. and Yu X. (1995): *High pressure air compressor valve fault diagnosis using feedforward neural networks.* — Mech. Syst. Signal Process., Vol.9, No.5, pp.527–536.

Jang J.S.R. and Sun C.T. (1993): *Functional equivalence between radial basis function networks and fuzzy systems.* — IEEE Trans. Neural Networks, Vol.4, No.1, pp.156–158.

Jang J.S.R. and Sun C.T. (1995): *Neuro-fuzzy modelling and control.* — Proc. IEEE, Vol.83, No.3, pp.378–406.

Kaymak U. (1998): *Fuzzy Decision Making with Control Applications.* — Ph.D. Th., Delft University of Technology, the Netherlands.

Keenan M.B. (1998): *Fine-Grained Object-Oriented Artificial Evolution of Artificial Neural Networks.* — Ph.D. Th., University of Alabama, Birmingham, USA.

Kiupel N., Köppen-Seliger B., Schulte Kellinghaus H. and Frank P.M. (1995): *Fuzzy residual evaluation concept (FREC).* — Proc. IEEE/SMC Conf., Vancouver, Canada, pp.13–18.

Klotzek P., Dalton T. and Frank P.M. (1998): *Application of sensitivity theory to fuzzy logic based FDI.* — Proc. INCO-Copernicus Workshop IQ^{2}FD, Kazimierz, Poland, pp.58–67.

Korba P. and Frank P.M. (1997): *Fuzzy control of non-linear systems using the Takagi-Sugeno model.* — Proc. TEMPUS PROJECT Symp. *System Modelling, Fault Diagnosis, Fuzzy Logic and Control*, Miskolc, Hungary.

Korbicz J., Obuchowicz A. and Patan K. (1998): *Network of dynamic neurons in fault detection systems.* — IEEE Int. Conf. *Systems, Man and Cybernetics*, San Diego, USA, (published on CD-ROM, No.98CH36218).

Kovács S., Kóczy L.T. and Bikfalvi P. (1998) *Application of an interpolation-based fuzzy logic controller in path tracking and collision avoidance of a vehicle.* — Workshop *European Scientific & Industrial Collaboration on Promoting Advanced Technologies in Manufacturing WESIC'98*, Girona, Spain.

Kruse R., Gebhardt J. and Klawonn F. (1994): *Foundations of Fuzzy Systems.* — Chichester: Wiley.

Kuipel N. and Frank P.M. (1997): *A fuzzy FDI decision-making system for the support of the human operator.* — Proc. IFAC Symp. *SAFEPROCESS'97*, Hull, UK, Pergamon Press, pp.721–726.

Kuipers B.J. (1994): *Qualitative Reasoning.* — Cambridge: MIT Press.

Kuipers B.J., Chiu C., Dalle Molle D.T. and Throop D.R. (1991): *Higher-order derivative constraint in qualitative simulation.* — Artif. Intell., Vol.51, pp.343–379.

Lane S.H., Handelman D.A. and Gelfand J.J. (1992): *Theory and development of higher-order CMAC neural-networks.* — Contr. Syst. Mag., Vol.12, No.2, pp.23–30.

Leonard J.A. and Kramer M.A. (1993): *Diagnosing dynamic faults using modular neural-nets.* — IEEE Expert Syst. Mag., Vol.8, No.2, pp.44–53.

Leonhardt S., Ludwig C. and Schwarz R. (1995): *Real time supervision for diesel engine injection.* — Contr. Eng. Practice, Vol.3, No.5., pp.1003–1010.

Leitch R. (1993): *Engineering diagnosis: Match problems to solutions.* — Proc. Int. Conf. *Fault Diagnosis: TOOLDIAG'93*, Toulouse, France, pp.837–844.

Levin R. and Narendra K.S. (1996): *Control of non-linear dynamical systems using neural networks: Part II: Observability, identification and control.* — IEEE Trans. Neural Networks, Vol.7, No.1, pp.34–42.

Li S. and Elbestawi M.A. (1996): *Fuzzy clustering for automated tool condition monitoring in machining.* — Mech. Syst. Signal Process., Vol.10, No.5, pp.533–550.

Lin C.T. and Lee C.S.G. (1991): *Neural network based fuzzy logic control and decision system.* — IEEE Trans. Comput., Vol.40, pp.1320–1336.

Lin C.T. and Wang H.P. (1993): *Classification of autoregressive spectral estimated signal patterns using an adaptive resonance theory neural network.* — Computers in Industry, Vol.22, No.2., pp.143–157.

Linkens D.A. and Abbod M.F. (1993): *Supervisory intelligent control using fuzzy logic hierarchy.* — Trans. Inst. M.C., Vol.15, No.3, pp.112–132.

Lopez-Toribio C.J., Patton R.J. and Daley S. (1998): *Fault-tolerant traction system control using fuzzy inference modelling.* — Proc. IFAC Workshop, *On-Line Fault Detection and Supervision in the Chemical Process Industries*, Lyon, France, Vol.I.

Lopez-Toribio C.J., Patton R.J. and Daley S. (1999): *Takagi-Sugeno fuzzy fault tolerant control of an induction motor.* — Special Issue of Neuro-Computing and Applications J. N-F Syst., (to appear, Dec., 1999).

Lunze J. (1994): *Qualitative modelling of linear dynamical systems with quantized state measurements.* — Automatica, Vol.30, No.3, pp.417–431.

Lunze J. and Schiller F. (1999): *An example of fault diagnosis by means of probabilistic reasoning.* — Contr. Eng. Practice, Vol.7, No.2, pp.271–278.

Lunze J., Schröder J. (1999): *Application of qualitative observation and prediction to a neutralisation process.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.I, pp.49–54.

Mamdani E. (1974): *Applications of fuzzy algorithms for control of simple dynamic plant.* — Proc. IEE, Vol.121, No.F, pp.1585–1588

Mamdani E. (1976): *Advances in the linguistic synthesis of fuzzy controllers.* — Int. J. Man-Machine Studies, Vol.8, pp.669–678.

Mamdani E. and Assilian S. (1975): *An experiment in linguistic synthesis with fuzzy logic controller.* — Int. J. Man-Machine Studies, Vol.7, No.1, pp.1–13.

Marseguerra M., Ricotti M.E. and Zio E. (1996): *Neural network-based fault detection in a pressurised water reactor pressuriser.* — Nucl. Sci. Eng., Vol.124, No.2, pp.339–348.

Moody J. and Darken C. (1989): *Fast learning in networks of locally-tuned processing units.* — Neural Computation, Vol.1, No.F, pp.281–294.

Naidu S., Zafirou E. and McAvoy T.J. (1990): *Use of neural-networks for failure detection in a control system.* — IEEE Contr. Syst. Magazine, Vol.10, pp.49–55.

Narendra K.S. (1998): *Neural networks for identification and control.* — Workshop Notes at IEEE CDC, Tampa, Fl (available from the Center for Systems Science, Yale University).

Narendra K.S. and Parthasarathy K. (1990): *Identification and control of dynamic systems using neural networks.* — IEEE Trans. Neural Networks, Vol.1, No.1, pp.4–27.

Nauck D. and Kruse R. (1998): *NEFCLASS-X-A soft computing tool to build readable fuzzy classifiers.* — BT Technology J., Vol.16, No.3, pp.180–190.

Nürnberger A., Nauck D. and Kruse R. (1997): *Neuro-fuzzy control based on the NEFCON model under MATLAB/SIMULINK.* — Proc. 2nd Online World Conf. *Soft Computing (WSC2)*, London: Springer-Verlag.

Obuchowicz A. and Korbicz J. (1998): *Evolutionary search with soft selection and forced direction of mutation.* — Proc. 7th Int. Symp. *Intelligent Information Systems*, Malbork, Poland, pp.300–309.

Obuchowicz A. and Patan K. (1997): *An algorithm of evolutionary search with soft selection for training multilayer feedforward neural networks.* — Proc. 3rd Conf. *Neural Network and Their Applications*, Kule, Poland, pp.123–128.

Pan M-C., Van Brussel H. and Sas P. (1998): *Intelligent joint fault diagnosis of industrial robots.* — Mech. Syst. Signal Process., Vol.12, No.4, pp.571–588.

Patan K. and Korbicz J. (1996): *Application of dynamic neural network in modelling and identification.* — Proc. 3rd Conf. *Methods and Models in Automation and Robotics*, Międzyzdroje, Poland, Vol.3, pp.1141–1146.

Patton R.J. (1997): *Robustness in model-based fault-diagnosis: The 1997 situation.* — A Rev. Contr., Vol.21, pp.103–123.

Patton R.J. (1999): Preface to the papers from the 3rd IFAC Symp. *SAFEPROCESS 97*, Contr. Eng. Practice, Vol.7, pp.201–202.

Patton R.J. and Chen J. (1994): *Review of parity space approaches to fault diagnosis for aerospace systems.* — J. Guid. Dyn. Contr., Vol.17, No.2, pp.278–285.

Patton R.J. and Chen J. (1996): *Neural networks in nonlinear dynamic systems fault diagnosis.* — Eng. Simulation, Vol.3, No.6, pp.905–925, (Special Issue *Engineering Diagnostics*).

Patton R.J., Chen J. and Liu G.P. (1997): *Robust fault detection of dynamic systems via genetic algorithms.* — Proc. I. Mech. E. Part I-J. Syst. and Contr. Eng., Vol.211, No.5, pp.357–364.

Patton R.J., Chen J. and Lopez-Toribio C.J. (1998): *Fuzzy observers for non-linear dynamic systems fault diagnosis.* — Proc. 37th IEEE Conf. *Decision and Control*, Tampa, Florida, pp.84–89.

Patton R.J., Chen J. and Siew T.M. (1994): *Fault diagnosis in non-linear dynamic systems via neural-networks.* — Proc. *Control'94*, Coventry, UK, Vol.2, pp.1346–1351.

Patton R.J., Frank P.M. and Clark R.N. (1989): *Fault Diagnosis in Dynamic Systems: Theory and Application.* — Contr. Eng. Series., New York: Prentice Hall.

Ren X. and Chen J. (1999): *A modified neural network for dynamical system identification and control.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.J, pp.463–468.

Roger J.S. (1993): *ANFIS: Adaptive-network-based fuzzy inference systems.* — IEEE Trans. Syst., Man Cybern., Vol.23, pp.665–685.

Ruano A.E., Lima J.M., Azevedo A.B., Duarte N.M. and Fleming P.J. (1999): *Automatic tuning of controllers using neural-genetic systems.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.K, pp.7–12.

Sard A. and Weintraub S. (1971): *A Book of Splines.* — New York: Wiley.

Schneider H. and Frank P.M. (1994): *Fuzzy logic based threshold adaptation for fault detection in robots.* — Proc. 3rd IEEE Conf. *Control Applications*, Glasgow, Scotland, pp.1127–1132.

Schneider H. and Frank P.M. (1996): *Observer-based supervision and fault detection in robots using non-linear and fuzzy-logic residual evaluation.* — IEEE Trans. Contr. Syst. Techn., Vol.4, No.3, pp.274–282.

Schumaker L.L. (1981): *Spline Functions: Basic Theory.* — New York: Wiley.

Sharif M.A. and Grosvenor R.I. (1998): *Process plant condition monitoring and fault diagnosis.* — J. Process Mech. Eng., Vol.212, No.1, pp.13–30.

Shen Q. and Leitch R. (1993): *Fuzzy Qualitative Simulation.* — IEEE Trans. Syst. Man Cybern., Vol.SMC-23, No.4, pp.1038–1061.

Simpson P.K. (1992): *Fuzzy min-max neural networks — Part1: Classification.* — IEEE Trans. Neural Networks, Vol.3, pp.776–786.

Simpson P.K. (1993): *Fuzzy min-max neural networks — Part 2: Clustering.* — IEEE Trans. Fuzzy Syst., Vol.1, pp.32–45.

Sousa J.M. (1998): *A Fuzzy Approach to Model Based Control.* — Ph.D. Th., Delft University of Technology, the Netherlands.

Sousa J.M., Setnes M. and Kaymak U. (1998): *Adaptive decision alternatives in fuzzy predictive control.* — Proc. IEEE World Congress *Computational Intelligence*, Anchorage, pp.698–703.

Stewart J.C. (1995): *The Application of Artificial Intelligence to Fault Detection in Hydraulic Cylinder Drive Systems.* — Ph.D. Th., Cardiff, Wales.

Takagi T. and Sugeno N. (1985): *Fuzzy identification of systems and its applications to modelling and control.* — IEEE Trans. Syst. Man Cybern., Vol.15, No.1, pp.116–132.

Tan Y. and Saif M. (1999): *Using neural networks for engine modelling and sensor failure diagnosis.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.Q, pp.203–208.

Tanaka K. and Sugeno N. (1992): *Stability analysis and design of fuzzy control systems.* — Fuzzy Sets Syst., Vol.2, No.45, pp.135–156.

Tanaka K., Takayuki I. and Wang H. (1996): *Design of fuzzy control systems based on relaxed LMI stability conditions.* — Proc. 35th CDC, Kobe, pp.598–603.

Tanaka K., Taniguchi T., Hua O. and Wang H. (1999): *Robust and optimal fuzzy control: A linear matrix inequality approach.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.K, pp.213–218.

Tzafestas S. (1989): *System fault diagnosis using the knowledge-based methodology,* In: Fault Diagnosis in Dynamic Systems: Theory and Application (Patton, Frank and Clark, Eds.). — New York: Prentice Hall, pp.509–572.

Vachkovv G. and Matsuyama H. (1992): *Fault diagnosis method by using fuzzy rule based models.* — Proc. 2nd Int. Conf. *Fuzzy Logic and Neural Networks*, Iizuka, Japan, pp.85-388.

Verbruggen H.B. and Babuka R. (1999): *Fuzzy Logic Control Advances in Applications.* — Singapore: Word Scientific Press.

Wang H., Brown M. and Harris C.J. (1994): *Fault detection for a class of unknown non-linear systems via associative memory networks.* — Proc. I. Mech. E., J. Syst. Contr. Eng., Vol.208, No.F, pp.101–107.

Wang H., Tanaka K. and Griffin M.F. (1995): *Parallel distributed compensation of a non-linear systems by Takagi and Sugeno fuzzy models.* — Proc. FUZZ-IEEE/IFES'95, New Orleans.

Wang L.X. and Mendel J. (1992): *Generating fuzzy rules by learning from examples.* — IEEE Trans. Syst. Man Cybern., Vol.22, No.F, pp.1414–1427.

Wasiewicz P. (1998): *Description of a sugar technology process.* — Institute of Automatic Control and Robotics, Warsaw University of Technology, Poland (internal report available from the authors of this paper).

Watton J. and Pham D.T. (1997): *An artificial neural network based approach to fault diagnosis and classification offluid power systems.* — J. Syst. Contr. Eng., Vol.211, No.4, pp.307–317.

Wilson D.J. (1998): *Neural networks for multivariate SPC.* — Ph.D. Th., Queens Univ. Belfast Faculty of Engineering, U.K.

Xiong Z., Wang X. and XU Y. (1999): *Non-linear system modelling using multiple neural networks.* — Proc. 14th IFAC World Congress, Beijing, China, Vol.N, No.143–198.

Yin C.M. (1993): *Application of artificial neural networks to condition monitoring.* — Ph.D. Th., Univ. Aberdeen, U.K.

Zadeh L.A. (1965): *Fuzzy sets.* — Inf. Contr., Vol.8, pp.338–353.

Zalzala A.M.S. and Fleming P.J. (1998): *Genetic Algorithms in Engineering Systems.* — London: Peter Peregrinus Press.

Zhuang Z. and Frank P.M. (1997): *Qualitative observer and its application to fault detection and isolation systems.* — Proc. I. Mech. E. Part., Int. J. Syst. Contr. Eng., Vol.211, No.4, pp.253–262.

Received: 15 March 1999
Revised: 13 September 1999