ONLINE FAULT DIAGNOSIS OF NONLINEAR SYSTEMS BASED ON NEUROFUZZY NETWORKS

H.T. Mok and C.W. Chan

Department of Mechanical Engineering,
The University of Hong Kong, Hong Kong

Abstract: Artificial intelligence techniques such as neural networks and fuzzy logic have been widely used in fault detection and diagnosis. Combining these two techniques, referred to as neurofuzzy networks, provides a powerful tool for modelling. B-spline neurofuzzy networks are used to model the residuals. The weights of the networks are trained online using recursive least squares method. Fuzzy rules are extracted from the networks and they provide linguistic description of the residuals. The qualitative information of the residuals facilitates isolation of the system faults. The proposed scheme is illustrated using a simulation example of a DC motor.

Keywords: fault diagnosis, neural networks, fuzzy logic, nonlinear systems, motor

1. INTRODUCTION

Fault detection and diagnosis of dynamic systems is important in engineering, as it helps to improve the reliability and safety of the systems. Complete reliance on human operators to cope with the situation when faults occurred has become increasingly difficult due to the high complexity of practical systems. To overcome the difficulties, automated fault monitoring procedures are developed.

During the last two decades, many investigations have been made using analytical approaches, based on quantitative models (Isermann, 1997). Residuals are generated to reflect inconsistencies between normal and faulty operations (Gertler, 1993). However, requirement for accurate analytical model implies any modelling error will affect the performance of the fault detection and diagnosis (Patton, *et al*., 2000). This is particularly true for nonlinear systems, which represent the majority of real processes.

Another approach to tackle this problem is to use qualitative model based on fuzzy logic (Fenton, *et al*., 2001) instead of quantitative model. In the fuzzy models, knowledge is expressed using fuzzy rules such as, IF-THEN. In this way, the requirement for accurate model can be relaxed (Venkatasubramanian, *et al*., 2003).

For monitoring purpose, the fault diagnostic scheme should be able to update its states so that any abnormal-changes can be captured. Neural networks is a strong candidate since it is able to be trained online and to approximate the nonlinear function with arbitrary accuracy (Frank and Köppen-Seliger, 1997). However, they do not give much insight into the behaviour of the system.

With the combination of fuzzy logic, which can express expert knowledge in linguistic form, a powerful tool for modelling nonlinear systems is formed and is referred to as neurofuzzy networks (Brown and Harris, 1994). Based on the neurofuzzy networks, expert knowledge can be included, extracted, and provides linguistic description of the faults. This facilitates isolation of system faults (Patton, *et al*., 2000b; Al-Jarrah and Al-Rousan, 2001).

Recently, a number of works have proposed the use of B-spline neurofuzzy network (BSNN) to implement the diagnostic system and effectiveness of BSNN in fault detection and isolation have been demonstrated (Patton, *et al*.,2000a; Uppal and Patton, 2000; Wang, *et al.*, 2001). However, there is no existing systematic method to use BSNN model for fault diagnosis.

In this paper, a system is first modelled by a BSNN with a set of training data. Another BSNN is then used to model the residuals online. Based on the residual model, a fuzzy rule base describing the residuals is extracted from the BSNN during operation. A coding procedure is then used to code the rule base into a simple pattern so that any fault can be detected readily and efficiently. For altered digits in the code, specific fuzzy rules corresponding to the symptoms are located and the fault can then be diagnosed.

The paper is organized as follows. A brief review of BSNN is presented in Section 2 and the fault diagnosis using BSNN is described in Section 3. The proposed scheme is illustrated by a simulation example of a DC motor given in Section 4.

2. B-SPLINE NEUROFUZZY NETWORK

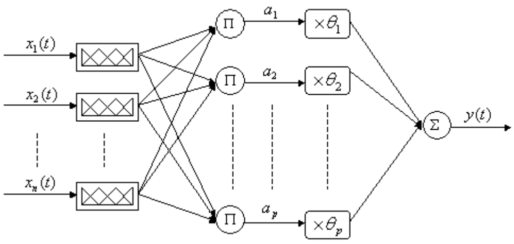

Fig. 1. B-spline neurofuzzy network.

The BSNN shown in Fig. 1 provides a useful link between neural networks and fuzzy systems and allows both approaches to be treated within a unified framework. It has the learning abilities of neural networks, which can approximate nonlinear functions with arbitrary accuracy, and the ability to incorporate fuzzy rules, which allows expert knowledge in linguistic form to be included.

An example of a fuzzy rule R_p in the BSNN:

R_p: IF $x_1(t)$ is negative big,
$x_2(t)$ is positive medium,
and …,
and $x_n(t)$ is positive small,
THEN $y(t)$ is negative medium.

Brown and Harris (1994) proposed to use B-spline basis functions as the membership functions of the fuzzified network input $x(t) = x = [x_1, \ldots, x_n]^T$, since they are compact and positive over the support $[\lambda_{j-\rho}, \lambda_j]$, i.e., $\mu_\rho^j(x) > 0$, for $x \in (\lambda_{j-\rho}, \lambda_j)$, and $\mu_\rho^j(x) = 0$, for $x \notin [\lambda_{j-\rho}, \lambda_j]$, and they form a partition of unity, i.e., $\sum_j \mu_\rho^j(x) \equiv 1$, $x \in [x_{\min}, x_{\max}]$, where λ is the knot value and μ_ρ^j is the j^{th} membership function of order ρ. A ρ^{th} order B-spline function is given by the following recurrent relationship:

$$\mu_\rho^j(x) = \left(\frac{x-\lambda_{j-\rho}}{\lambda_{j-1}-\lambda_{j-\rho}}\right)\mu_{\rho-1}^{j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-\rho+1}}\right)\mu_{\rho-1}^{j}(x) \quad (1)$$

$$\mu_1^j(x) = \begin{cases} 1 & if\ x \in [\lambda_{j-1}, \lambda_j) \\ 0 & otherwise \end{cases} \quad (2)$$

It is shown that under certain conditions (Brown and Harris, 1994), the output of BSNN is given by:

$$y(t) = a^T(x(t))\theta \quad (3)$$

where $\theta = [\theta_1, \theta_{2,\ldots,}\theta_p]^T$ is the network weight, $a(x(t)) = [a_1(x(t)), a_2(x(t)), \ldots, a_p(x(t))]^T$ is the transformed input, $a_i(x(t))$ is the tensor product of the univariate B-spline basis functions:

$$a_i(x(t)) = \prod_{k=1}^{n} \mu_\rho^j(x_k(t)) \quad (4)$$

and p is the number of weights in the network:

$$p = \prod_{i=1}^{n}(r_i + \rho_i) \quad (5)$$

where r is the number of inner knots in the partition of $x(t)$. As the fuzzy sets in the BSNN are distributed over the neighbourhood regions, the approximation of a nonlinear function by the BSNN is generally smooth.

The training of BSNN involves finding a set of weights θ that minimizes the cost function:

$$J = \frac{1}{n}\sum_{k=1}^{n}(e(k))^2 \quad (6)$$

where n is the number of training data, $e(k)$ is the error term between the target output and the network output. Since the network output is linear-in-the-weight of the network inputs, the well-known recursive least squares (RLS) estimator can be used to find the optimal set of weights θ online:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{P(t-1)a(x(t))[y(t) - a^T(x(t))\hat{\theta}(t-1)]}{1 + a^T(x(t))P(t-1)a(x(t))} \quad (7)$$

and the covariance matrix P is updated by:

$$P(t) = P(t-1) - \frac{P(t-1)a(x(t))a^T(x(t))P(t-1)}{1 + a^T(x(t))P(t-1)a(x(t))} \quad (8)$$

3. FAULT DIAGNOSIS USING BSNN

When a system is operating normally, there will be no difference between the output observed from the system and that predicted from the model. The difference, referred to as residual, will depart from zero when faults occur. However, it does not contain information of the nature of fault occurred. B-spline neurofuzzy networks are used to model the residual so that symbolic information of the fault can be obtained and this helps to diagnose the fault.

3.1 *Residual model*

In the BSNN, residual is the network output while any controllable input of the system can be chosen as the network input. Some parameters of the BSNN have to be selected before the training of the network. In this paper, as an example, a network with a second order (triangular) basis function and one inner knot is used to implement the residual model in both input and output partition spaces. This configuration divides the input and output partition space into three linguistic variables as shown in Fig. 2. The input variables S, M, and B denotes respectively small, medium, and big while the output variables N, Z, and P denotes respectively negative, zero, and positive.

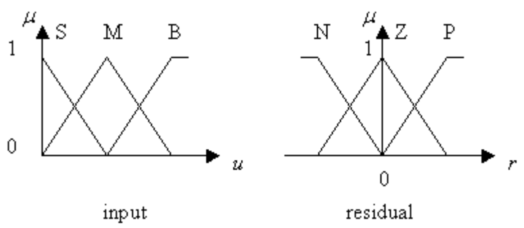

Fig. 2. Input and output partition spaces of BSNN

3.2 *Fuzzy rules extraction*

With the presence of residual model, a set of optimal weights is obtained after a period of training. A fuzzy rule base can then be extracted from the weights and it provides linguistic descriptions of the residual and thus the fault.

Consider a fuzzy rule R_{ij} in the BSNN:

R_{ij}: If x is A_i, then y is B_j (c_{ij}),
for $i = 1,\ldots,p$ and $j = 1,\ldots,q$

where A_i, B_j denote respectively fuzzy sets in the input and output partition spaces, p is the input partition space number, q is the output partition space number, c_{ij} is the level of confidence of the rule R_{ij} being true. Brown and Harris (1994) showed that, given a set of optimal weights θ of the network, it is possible to find the equivalent fuzzy representation with the level of confidence given by:

$$c_{ij} = \mu_{B_j}(\theta_i) \tag{9}$$

Level of confidence of a rule is simply the B-spline basis function of the associated network weight. Figure 3 shows the fuzzy rule extraction from an optimal weight θ_i and the resulting rule base can be interpreted as follows:

(1)	If x is A_i, then y is B_1	(*0.0000*)
(2)	If x is A_i, then y is B_2	(c_{i2})
(3)	If x is A_i, then y is B_3	(c_{i3})

This feature allows the integration of both numerical data and symbolic knowledge within a single framework and therefore BSNN is a very powerful tool in fault diagnosis.

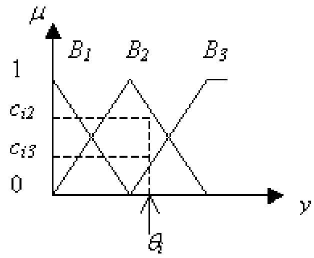

Fig. 3. Fuzzy rules extraction.

3.3 *Fault diagnosis*

Residual model is built using BSNN and a fuzzy rule base describing the residual is extracted from the BSNN. The next step is to use the fuzzy rule base to diagnose the fault.

For online monitoring purpose, the weights of the BSNN are updated recursively using the algorithm discussed in Section 2. Whenever a fuzzy rule base is obtained, it is compared with the one obtained under normal operating condition to see if there is any inconsistency. The discrepancy accounts for the symptom of fault. Faults are then classified based on the symptoms.

A systematic approach is proposed here to locate the difference. For each group of rules under the same antecedent, the one with the highest level of confidence is highlighted and the position is coded. For example, the second rule within the group is the one with the highest confidence, a digit of 2 will be given to the group as shown in Fig. 4.

Therefore, when a code obtained is different from the normal one, a fault is detected and the change of each particular digit helps to locate the problem occurred in the system more efficiently. When a complete code set is built for every single fault, the fault can then be classified readily with the fault code. For example, 2-2: normal, 1-1: fault 1, 3-3: fault 2, 1-3: fault 3… etc.

<u>Group 1</u>

1	If x is A_1, then y is B_1	0.0000
2	If x is A_1, then y is B_2	*1.0000*
3	If x is A_1, then y is B_3	0.0000

<u>Group 2</u>

1	If x is A_2, then y is B_1	*0.7020*
2	If x is A_2, then y is B_2	0.2980
3	If x is A_2, then y is B_3	0.0000

⋮

Code: 2-1-…

Fig. 4. Coding of fuzzy rule base

4. DC MOTOR EXAMPLE

Consider a DC motor with a shunt field circuit as shown in Fig. 5. The motor is described by a set of nonlinear ordinary differential equations (Watanabe, *et al.*, 1985):

$$\dot{i}_f = -\frac{R_f}{L_f} i_f + \frac{1}{L_f} U \tag{10}$$

$$\dot{i}_a = -\frac{R_a}{L_a} i_a + \frac{1}{L_a} U - \frac{I}{L_a} i_f \omega \tag{11}$$

$$\dot{\omega} = \frac{I}{J} i_f i_a - \frac{D}{J} \omega \tag{12}$$

where i_f is the field current, i_a the armature current, ω the rotational speed, U the input voltage, R_f and L_f the field resistance and inductance, R_a and L_a the armature resistance and inductance, I the mutual inductance between L_a and L_f, D the viscous resistance of the load and J the inertia moment of the load.

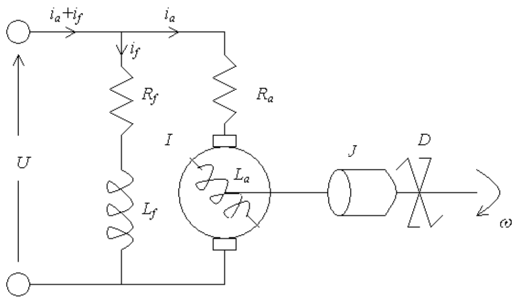

Fig. 5. DC motor with a shunt field circuit.

Table 1. Values of parameters of DC motor

Parameter	Value
R_f	50 Ω
L_f	20 H
R_a	3.8 Ω
L_a	0.5 H
D	0.042 $Nms\ rad^{-1}$
J	0.4 kgm^2
M	0.221 H

Let the state vector $[x_1\ x_2\ x_3]^T = [i_f\ i_a\ \omega]^T$, and the measurement output vector $[y_1\ y_2]^T = [i_f + i_a\ \omega]^T$. Substituting the parameter values in Table 1 into equation (10-12) yields:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -2.5 & 0 & 0 \\ 0 & -7.6 & 0 \\ 0 & 0 & -0.105 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0.05 \\ 2 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0.442\dot{x}_1\dot{x}_3 \\ 0.5525\dot{x}_1\dot{x}_2 \end{bmatrix} \tag{13}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{14}$$

The input voltage is given by $U = 10\sin(\pi t/15)+100V$ and the initial states, $x_1(0)$, $x_2(0)$, and $x_3(0)$ are all zeros. Two BSNNs, each with a second order basis function and one inner knot in both input and output partition spaces, are used to approximate the DC motor and the outputs are shown in Fig. 6.

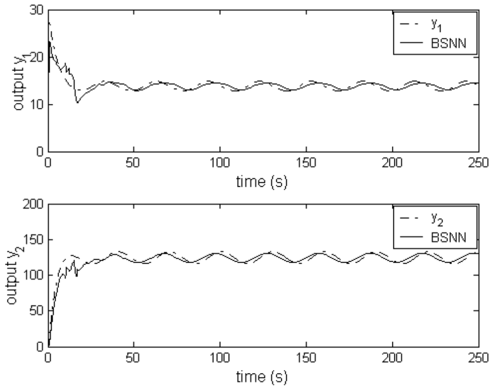

Fig. 6. Model outputs of DC motor.

A fault of 20% increase in parameter R_a, arising from a fault in the brush in the rectifier of the motor is introduced into the DC motor at t = 100s. Outputs of the motor under the faulty operation are shown in Fig. 7 and both the outputs y_1 and y_2 decrease when the fault occurs.

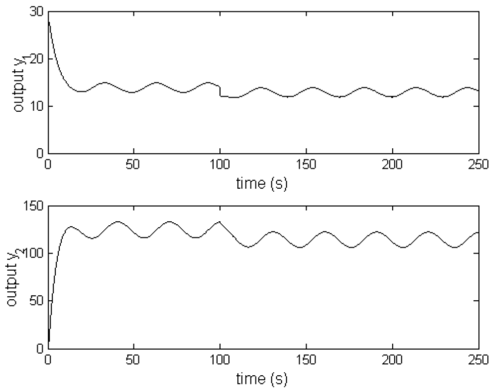

Fig. 7. Faulty outputs of DC motor.

4.1 Residual model

Two residuals r_1 and r_2 are obtained by the difference between the measured outputs and the model outputs. B-spline neurofuzzy networks are then used to model the residuals r_1 and r_2 with input voltage U as the network input so that symbolic information of the residuals can be obtained.

The two BSNNs are implemented with three triangular basis functions for U, r_1, and r_2. From (5), the total number of weights in the each network is 3. The B-spline fuzzy membership functions for input and output spaces are shown in Fig. 8. The initial weights and the covariance matrix of the RLS estimator are set to 0 and **I** respectively. The residual models and the adaptations of the network weights for 250s are shown in Fig. 9-10 and it can be seen that the network weights converge after a period of training.

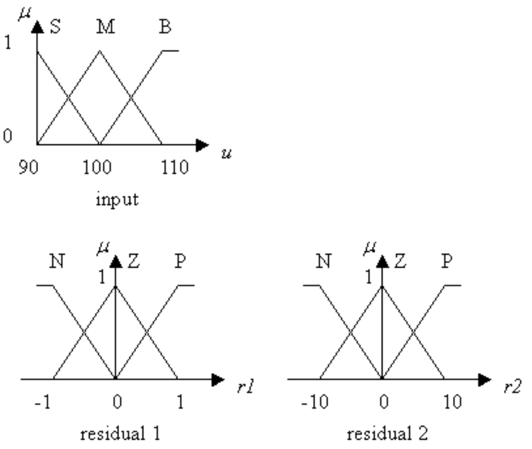

Fig. 8. Fuzzy sets of network input and outputs

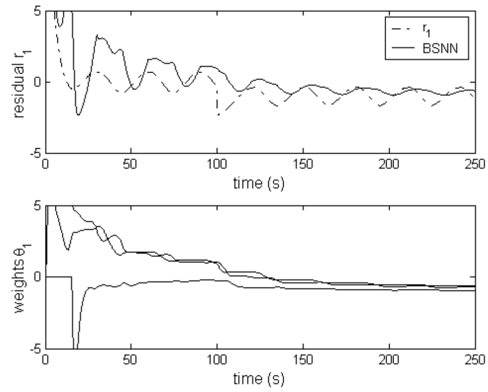

Fig. 9. Residual model and adaptation of network weights for r_1.

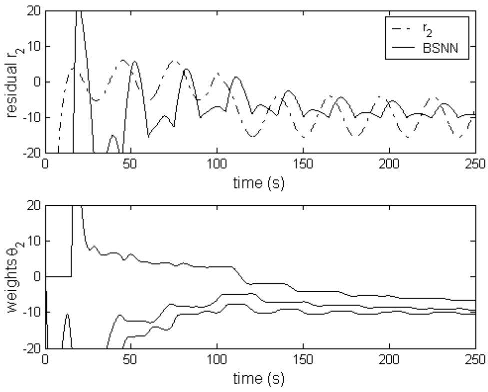

Fig. 10. Residual model and adaptation of network weights for r_2.

4.2 Fuzzy rules extraction

A rule base containing 18 fuzzy rules is extracted from the BSNNs. Table 2 shows the rule base under normal operating condition while Table 3 shows the rule base under faulty operating condition at t = 250s.

For each group of rules, the one with the largest confidence is highlighted with italic font and the position is coded. Obviously, under normal operation the middle rules in each group with “r is Z” as rule consequent are highlighted as residual always equals zero. However, this pattern changes from 2-2-2-2-2-2 to 1-1-1-1-1-1 when a fault occurs as shown in Table 4. Both the r_1 and r_2 are shifted from zeros (Z) to negative (N) values. Since a pattern different from the normal one is observed, a fault is detected. The next step is to diagnose the fault using classification techniques.

Table 2. Rule base under normal condition

No.	Rule	Confidence
1	If u is S, then r1 is N	0.0001
2	*If u is S, then r1 is Z*	*0.9999*
3	If u is S, then r1 is P	0.0000
4	If u is M, then r1 is N	0.0000
5	*If u is M, then r1 is Z*	*1.0000*
6	If u is M, then r1 is P	0.0000
7	If u is B, then r1 is N	0.0000
8	*If u is B, then r1 is Z*	*1.0000*
9	If u is B, then r1 is P	0.0000
10	If u is S, then r2 is N	0.0000
11	*If u is S, then r2 is Z*	*0.9996*
12	If u is S, then r2 is P	0.0004
13	If u is M, then r2 is N	0.0008
14	*If u is M, then r2 is Z*	*0.9992*
15	If u is M, then r2 is P	0.0000
16	If u is B, then r2 is N	0.0000
17	*If u is B, then r2 is Z*	*0.9999*
18	If u is B, then r2 is P	0.0001
	Code: 2-2-2-2-2-2	

Table 3. Rule base under faulty condition

No.	Rule	Confidence
1	*If u is S, then r1 is N*	*0.9406*
2	If u is S, then r1 is Z	0.0594
3	If u is S, then r1 is P	0.0000
4	*If u is M, then r1 is N*	*0.7035*
5	If u is M, then r1 is Z	0.2965
6	If u is M, then r1 is P	0.0000
7	*If u is B, then r1 is N*	*0.6257*
8	If u is B, then r1 is Z	0.3743
9	If u is B, then r1 is P	0.0000
10	*If u is S, then r2 is N*	*0.6627*
11	If u is S, then r2 is Z	0.3373
12	If u is S, then r2 is P	0.0000
13	*If u is M, then r2 is N*	*1.0000*
14	If u is M, then r2 is Z	0.0000
15	If u is M, then r2 is P	0.0000
16	*If u is B, then r2 is N*	*0.9305*
17	If u is B, then r2 is Z	0.0695
18	If u is B, then r2 is P	0.0000
	Code: 1-1-1-1-1-1	

4.3 Fault diagnosis

From the faulty rule base, it is observed that both r_1 and r_2 become negative (N) and this accounts for the symptoms of the fault. With some expert knowledge shown in Table 4, the fault can be isolated readily using the symptoms. (For validity of Table 4, the magnitude of the each fault is 20% deviated from its normal value.)

Table 4. Mapping from symptoms to fault

r_1	r_2	Fault
P	Z	$R_f\uparrow$
N	Z	$M\downarrow$
P	N	$D\uparrow$
N	N	$R_a\uparrow$

In this example, the fault can be simply classified as an increase in the system parameter R_a using Table 4. This may be due to a fault in the brush in the rectifier of the motor. However, when the system is highly nonlinear with the input, the complete fault code can be utilized for classification purpose.

5. CONCLUSION

An intelligent fault diagnosis scheme based on B-spline neurofuzzy networks is proposed for monitoring nonlinear systems. A residual model is first built using the BSNN. The weights of the network are then trained online using recursive least squares algorithm for monitoring purpose. Fuzzy rules are extracted from the network and the rule base is coded. The discrepancy from the normal rule base accounts for the heuristic symptoms, which facilitates the fault diagnosis of the system. The proposed scheme is illustrated using a simulation example of a DC motor system.

6. ACKNOWLEDGEMENTS

This work was supported by the HKSAR RGC Grant (HKU 2050/02E) and The University of Hong Kong.

REFERENCES

Al-Jarrah, O.M. and M. Al-Rousan (2001). Fault detection and accommodation in dynamic systems using adaptive neurofuzzy systems. *IEE Proceedings - Control Theory and Applications*, **148(4)**, pp. 283-290.

Brown M. and C.J. Harris (1994). *Neurofuzzy adaptive modelling and control*. Prentice Hall, New York.

Fenton, W.G., T.M. McGinnity and L.P. Maguire (2001). Fault diagnosis of electronic systems using intelligent techniques: A review. *IEEE Transactions on Systems Man And Cybernetics Part C - Applications And Reviews*, **31(3)**, pp. 269-281.

Frank, P.M. and B. Köppen-Seliger (1997). Fuzzy logic and neural network applications to fault diagnosis. *International Journal of Approximate Reasoning*, **16(1)**, pp. 67-88.

Gertler, J. (1993). Residual generation in model-based fault-diagnosis. *Control-Theory and Advanced Technology*, **9(1)**, pp. 259-285.

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods - An introduction. *Control Engineering Practice*, **5(5)**, pp. 639-652.

Patton, R.J., J. Chen and H. Benkhedda (2000a). A study on neuro-fuzzy systems for fault diagnosis. *International Journal of Systems Science*, **31(11)**, pp. 1441-1448.

Patton, R.J., F.J. Uppal and C.J. Lopez-Toribio (2000b). Soft computing approaches to fault diagnosis for dynamic systems: A survey. *Proceeding of IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes 2000*, Budapest, Hungary, vol. 1, pp. 303-315.

Uppal, F.J. and R.J. Patton (2000). Application of B-spline neuro-fuzzy networks to identification, fault detection and isolation. *UKACC International Conference on Control 2000*, Cambridge, UK.

Venkatasubramanian V., R. Rengaswamy and S.N. Kavuri (2003). A review of process fault detection and diagnosis Part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, **27**, pp. 313-326.

Wang, Y., C.W. Chan and K.C. Cheung (2001). Intelligent fault diagnosis based on neurofuzzy networks for nonlinear dynamic systems. *Proceeding of IFAC Conference on New Technologies for Computer Control 2001*, Hong Kong, China, pp. 101-104.

Watanabe, K., M. Sasaki and D.M. Himmelblau (1985). Determination of optimal measuring sites for fault-detection of nonlinear-systems. *International Journal of Systems Science*, **16(11)**, pp. 1345-1363.