Fault diagnosis of a hydraulic actuator circuit using neural networks—an output vector space classification approach

W J Crowther[1], **K A Edge**[1], **C R Burrows**[1], **R M Atkinson**[2] and **D J Woollons**[2]
[1] Fluid Power Centre, University of Bath
[2] School of Engineering, University of Exeter

Abstract: This paper presents a neural network approach to fault diagnosis of dynamic engineering systems based on the classification of surfaces in system output vector space. A simple second-order system is used to illustrate graphically the nature of the diagnosis problem and to develop theory. The approach is then applied to the diagnosis of a laboratory-based hydraulic actuator circuit. Results are presented for networks trained on both simulation and experimental data. An important achievement is the diagnosis of experimental faults using a network trained only on simulation data.

Keywords: fault diagnosis, neural networks, hydraulic systems

NOTATION

$\mathbf{A}$	system matrix
A_p	piston area (m^2)
$\mathbf{B}$	input matrix
B_f	fluid bulk modulus (Pa)
c	damping coefficient (N s/m)
$\mathbf{C}$	output matrix
$\mathbf{D}$	feedforward matrix
i_v	valve current (A)
k	spring stiffness (N/m)
k_f	dynamic friction coefficient (N/m s)
k_l	leakage coefficient (l/min bar)
k_v	valve flow coefficient ($m^3/s\ Pa^{1/2}$)
M	mass (kg)
P_p	piston pressure (bar)
$\dot{P}_p$	rate of change of piston pressure (bar/s)
P_s	supply pressure (bar)
ΔP	cylinder differential pressure (bar)
q	flow into cylinder (m^3/s)
q_l	leakage flow (m^3/s)
u	position input (m)
V	trapped volume (m^3)
x	position (m)
$\boldsymbol{x}$	state vector
x_e	spring extension (m)
x_i	displacement of end of spring (input)
x_v	spool relative displacement
$\dot{x}$	velocity (m/s)
$\ddot{x}$	acceleration (m/s^2)
$\boldsymbol{y}$	output vector

The MS was received on 19 February 1997 and was accepted for publication on 29 August 1997.

1 INTRODUCTION

With increased demand for availability in fluid power system applications, it is becoming increasingly cost effective to specify in-built diagnostics as part of the system design. Diagnostic information can be used as an aid to manual fault finding or as part of a wider architecture for system monitoring and control. The task of a diagnostic system is to use data from one or more sensors to establish information regarding the fault condition of a plant (Fig. 1). This task is non-trivial in most cases since it is usually physically impossible (and financially unviable) to provide separate sensors for each possible fault.

A number of approaches have been developed for diagnosis of engineering systems that fit within the model shown in Fig. 1. These may be grouped under the broad headings of:

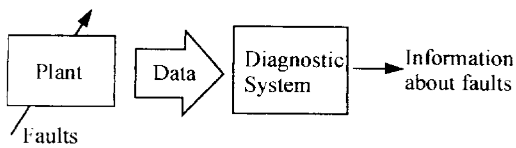


Fig. 1 The diagnostic process

1. Expert systems (**1**–**3**)
2. Qualitative reasoning (**4**, **5**)
3. Model-based diagnosis from an artificial intelligence perspective (**6**–**11**)
4. Model-based diagnosis from control engineering (**12**–**15**)
5. Neural networks (**16**–**21**)

Hybrid applications are common, e.g. the use of neural networks to classify residuals from model-based schemes (**22**) and the use of an expert system to post-process results from a neural network (**23**, **24**).

Fluid power systems present two main challenges to building diagnostic architectures. Firstly, the systems tend to be very non-linear and, secondly, behaviour is frequently dynamic, meaning that the output of the system is a function of time as well as any inputs to the system.

In an expert systems approach, problem non-linearity results in an increase in the number and complexity of rules required for diagnosis. Similarly, for qualitative reasoning approaches, non-linearity increases the size of the search space and results in slower and less reliable diagnosis. Model-based approaches in general typically rely on linear models, so increasing system non-linearity leads to larger modelling errors and less accurate diagnosis.

While qualitative reasoning shows some promise for the diagnosis of truly dynamic systems, artificial intelligence techniques in general are not well suited to reasoning about dynamic behaviour. In contrast, model-based techniques from control engineering make use of well-established numerical modelling techniques and are relatively successful.

The case for using neural networks to solve difficult, so-called 'intelligent' problems has perhaps been overstated in the last few years. However, for the present problem, neural networks offer a number of advantages: in particular, their ability to deal with highly non-linear dynamic systems and high speed of operation when implemented in hardware. This paper focuses on a neural network approach to diagnosis based on the classification of patterns in system output vector space. A novel aspect of the work is that a physical understanding of the system to be diagnosed is used to determine the inputs required for the neural network. This is in contrast to approaches where all available data are presented to the network in the expectation that the relevant information can somehow be extracted.

The following section provides a brief introduction to neural networks. This is followed by a simple example illustrating how diagnosis of dynamic systems can be conceptualized in terms of pattern classification. Sections 4, 5 and 6 represent the main body of work in this paper and describe the application of neural networks to diagnosis of a laboratory-based hydraulic actuator system. A summary of the approach is given in Section 7 and conclusions are drawn in Section 8.

2 NEURAL NETWORKS

The fundamental premise in neural computing is that a complex task can be performed by a large number of simple, interconnected processing elements in a network. In neural computing, in contrast to traditional computing, networks are *trained*, rather than programmed. Thus, instead of typing in a list of instructions to form a program, a neural network (NN) is given a set of input/output examples that it has to learn. A good introduction to the subject is given by Bishop (**25**) and Rumelhart *et al.* (**26**).

The issue of machine learning has been at the heart of neural network research since its beginnings in the late 1940s (**27**). However, the major breakthrough which fuelled the resurgence of interest in the mid-1980s was the development of an efficient learning algorithm for multilayer networks with non-linear processing elements (**28**). Also of importance was the proof that a network with at least one hidden layer containing non-linear processing elements could learn any continuous, single-valued, bounded function (**29**).

Of the different types of neural network available, the multilayer perceptron (MLP) trained using a variation of the error back-propagation algorithm is the most widely applied and is the type of network used in the present work.

3 SIMPLE DIAGNOSTIC EXAMPLE

3.1 Background

To gain an understanding of the problems involved in diagnosing dynamic engineering systems, a simple second-order mass–spring–damper system was considered as the basis of a series of numerical experiments:

$$M\ddot{x} + c\dot{x} + kx = kx_i \tag{1}$$

This can be expressed in state variable form as

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}x_i \tag{2}$$

with

$$\boldsymbol{x} = [x \;\; \dot{x}]^{\mathrm{T}}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k/M & -c/M \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ k/M \end{bmatrix}$$

and the output vector, $\boldsymbol{y}$, can be expressed by

$$\boldsymbol{y} = \mathbf{C}\boldsymbol{x} + \mathbf{D}x_i \tag{3}$$

with

$$\boldsymbol{y} = [x_e \ \dot{x} \ \ddot{x}]^T$$

$$\mathbf{C} = \begin{bmatrix} k & 0 \\ 0 & 1 \\ -k/M & -c/M \end{bmatrix}$$

and

$$\mathbf{D} = [-k \ 0 \ k/M]^T$$

3.2 Simulation results

If the three variables describing the system output vector in the example above are used as orthogonal axes, then it is possible to visualize the behaviour of the system in output vector space using a three-dimensional plot. Using this technique, the trajectories for the mass–spring–damper system with two non-zero initial conditions are shown in Fig. 2a. In this case parameters have been chosen such that the system is underdamped and thus undergoes a number of oscillations before reaching steady state (the centre of the spiral). If the same system is now excited by a time-varying random input and the output vector sampled at periodic intervals, a series of points lying on a plane in output vector space is generated (Fig. 2b). Note that increasing the level of excitation in the input signal increases the size of the output vector space occupied by the system.

Consider now a system with 'normal' parameter values: $M = 1$ kg, $c = 2$ N s/m and $k = 1$ N/m, i.e. a critically damped system with a natural frequency of 1 rad/s. Faults occurring due to changes in stiffness, k, or damping, c, can now be visualized in output vector space by plotting surfaces associated with discrete values of the faulty parameters, as shown in Fig. 3. For simultaneous faults (changes in both parameters), similar surfaces at different orientations are obtained.

3.3 Fault diagnosis

Fault diagnosis may be thought of in terms of parameter estimation (i.e. given the data shown in Fig. 3, determine the values of the parameters c and k in the underlying model) or in terms of classification (i.e. given an arbitrary surface in output vector space, to which of the surfaces shown in Fig. 3 is it closest?). Whichever point of view is taken, it is possible to solve the problem using statistical techniques or by using neural networks (which are essentially parallel implementations of statistical algorithms).

3.4 System excitation

The issue of system excitation is crucial to successful fault diagnosis of dynamic systems. If the input signal is steady (zero excitation) then the system occupies a single point in output vector space and both estimation and classification methods relying on distributed data will fail. For weakly excited systems, diagnosis is still possible, but the problem is numerically ill conditioned. This is reflected in unacceptably long training times for neural networks.

3.5 Neural network implementation

In the present example, the task of a neural network is to learn the mapping between the behaviour of the system in output vector space and the values of parameters c and k in the underlying model. During network training, vectors representing the system output variables are used as the training inputs and the correspond-

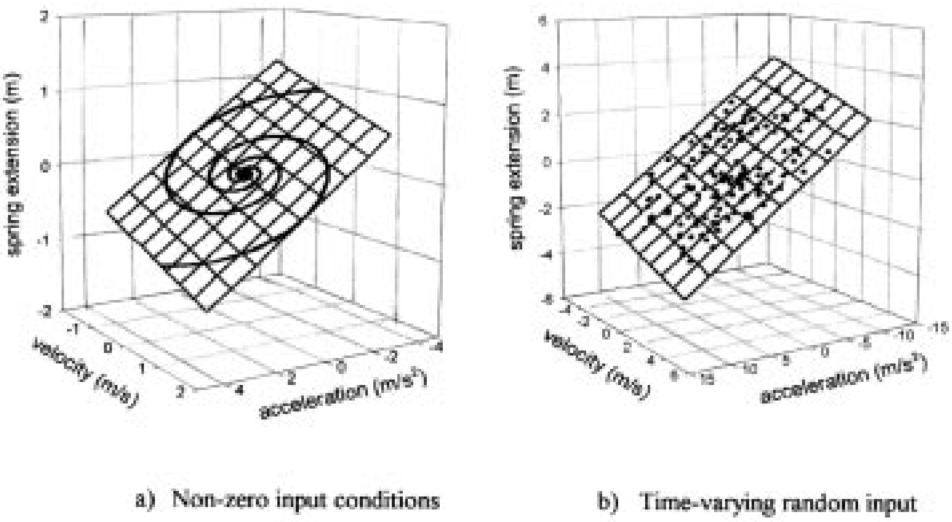
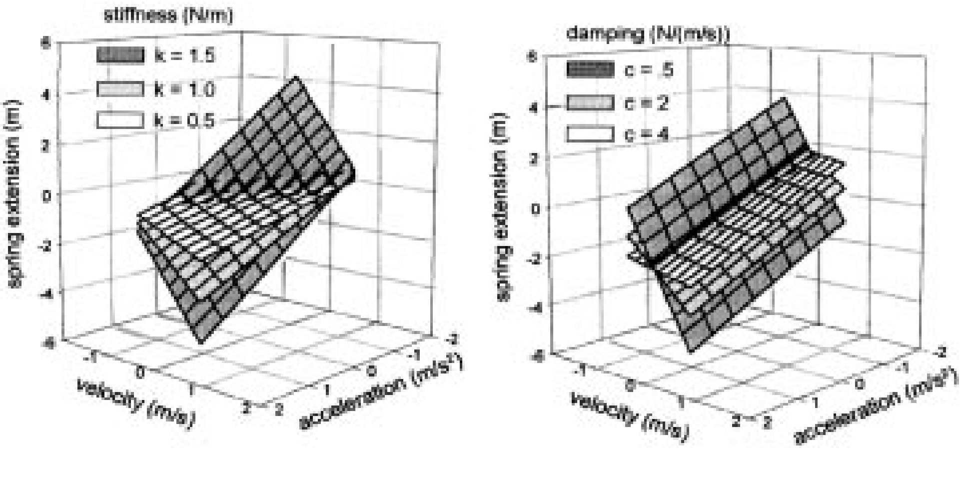


Fig. 2 Output vector space representation of the mass–spring–damper system



(a) k varying; c constant (=2.0)   (b) c varying; k constant (=1.0)

Fig. 3 Visualization of faults as surfaces in output vector space

ing system parameter vectors are used as the training targets. Once the network has been trained it can be queried with output vector space variables from a potentially faulty system whose parameter values are unknown. If the training process has been successful, the network will output a vector indicating the values of c and k, and hence the fault status of the system.

Assuming an arbitrary sample size of 10 points in three-dimensional output vector space and a separate output neuron for each fault, a neural network with 30 input neurons and 2 output neurons is required for fault classification (Fig. 4). Assuming a standard three-layer network, the number of neurons required in the hidden layer depends on the required accuracy of the diagnosis and can only be found by experiment (**30**). Networks were implemented using commercial software running on a Windows P120 MHz PC with 32 Mb of RAM.

3.6 Network training

By experimentation, it was demonstrated that network training time is reduced by increasing the size of the sample window and by increasing the degree of system excitation. For the data structure shown in Fig. 4, single faults could be learnt to a high degree of accuracy in approximately 1 min. However, even for this simple case, a network trained with combinations of faults resulted in a tenfold increase in training time (**31**). With more complex systems, the number of potential fault combinations rapidly creates an unmanageable quantity of training data. As a consequence, it is inappropriate to use combinations of faults for network training.

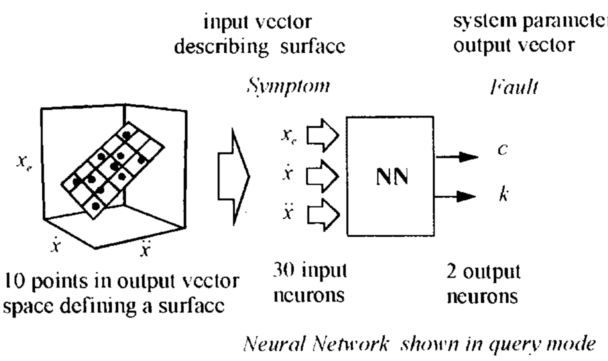


Fig. 4 Neural network structure for diagnosis of the mass–spring–damper system

4 EXPERIMENTAL HYDRAULIC ACTUATOR SYSTEM

4.1 Circuit description

A schematic of the test rig used to validate ideas developed in the previous section is shown in Fig. 5. In this rig, a 46 kg trolley-mounted mass is moved by a 0.1 m stroke actuator controlled by an electrically operated servo valve. The trolley position is measured with a displacement transducer and this signal is used to close a positional control loop. Leakage between the annulus and piston sides of the actuator can be introduced by opening a cross-line bleed valve. A load is applied to the trolley through a second, passive actuator. The dynamic friction characteristics of the load can be modified by adjusting the bleed valve that connects the two sides of the actuator.

Existing sensors on the rig allow measurement of the servo valve current, i, the differential pressure between the piston and annulus sides of the drive actuator, ΔP, and the trolley position, x. Data were recorded using a PC-based acquisition system.

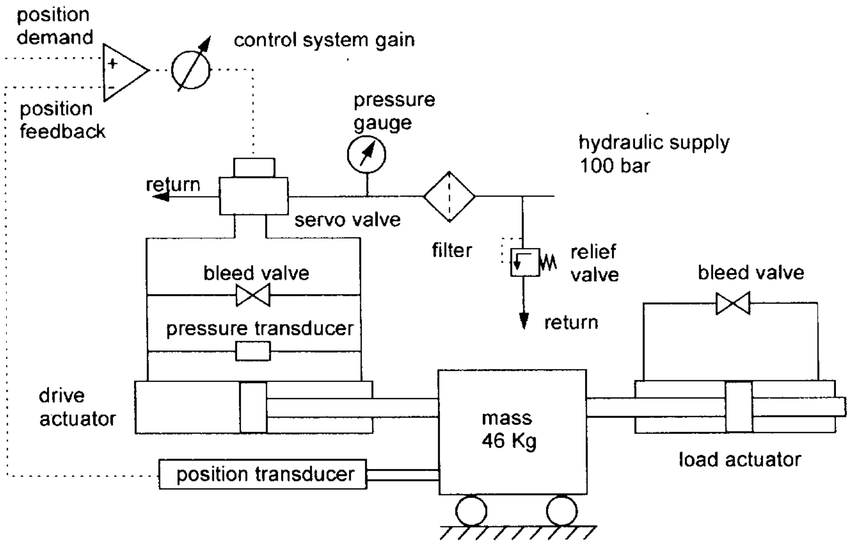


Fig. 5 Experimental actuator rig

4.2 Faults of interest

In this study, it was decided to focus on diagnosing the following faults:

(a) incorrect supply pressure,
(b) increased drive actuator cross-line leakage,
(c) increased load dynamic friction.

These faults are relatively common in practice and can be introduced into the rig without causing damage.

When more than one fault can occur in a system, it is important to define the conditions under which diagnosis is to be achieved. The simplest case is that of a single fault occurring in a single-fault environment. This implies that the system parameter matrix changes along a unique vector. More complex is the case of single faults occurring in a system that can experience many faults. Here the system parameter matrix can change in a direction given by one of a number of fault vectors. The most complex case is that of simultaneous faults. Here the system parameter matrix can change in a direction given by any combination of the fault vectors.

For the present work, it is assumed that faults occur singly, but exist in a multiple-fault environment.

4.3 Choice of input signal

The hydraulic circuit shown in Fig. 5 could be used in a number of different applications, and each would have its own characteristic duty cycle. A particular problem occurs if the duty cycle contains long periods of little or no excitation, since during these it will be difficult to identify the faults of interest. In the present work, the problem has been avoided by driving the system with a persistently exciting position demand signal made up of a series of low-amplitude, random step inputs. The maximum peak-to-peak magnitude of the input was chosen to be 5 mm (10 per cent of the actuator stroke). Having fixed the amplitude, the signal switching time was determined on the arbitrary basis that the system should not be in steady state for more than approximately 10 per cent of the total time. This was achieved with a switching time of 20 ms.

The excitation problem for arbitrary duty cycles may be solved by continually assessing the level of system excitation and recording data only when the excitation exceeds a given threshold. However, this approach was not used in the work presented here.

A theoretically based discussion of issues relating to experiment design for system identification may be found in reference (**32**).

4.4 Model development

A model of the actuator rig was developed using Bath*fp*, a component-based, variable-time-step simulation environment developed at the University of Bath. Attention was focused on obtaining accurate servo valve and actuator models. All pipes were modelled as lumped-parameter elements with an associated compressible volume. The load actuator assembly and drive actuator bleed valve were not modelled explicitly. Instead, the effects of these components were included as parameters within the drive actuator model.

The process of manually adjusting model parameters to optimize simulation results with respect to experimental data is a difficult and tedious task. To automate this, a fast, fixed time step, transmission-line model simulation of the rig was developed (**33**) and the model parameters optimized using a genetic algorithm. The method is computationally intensive, but generally gives excellent results given sufficient time. Optimization typically took 80 000 generations and an elapsed time of around 12 h on a fast workstation. Parameter estimation using a conventional least-squares approach was not used because previous experience had shown very poor performance for other similar, highly non-linear hydraulic circuits.

The optimized parameter values were validated using test data not used in the optimization process (Fig. 6). Note, however, that the demand signal used for both optimization and validation is a randomly generated series of steps. Only in this respect are the validation data similar to the optimization data.

An important goal of the present work is to demonstrate the feasibility of using NNs trained on simulation data to diagnose faults in experimental data. In practice, it can be expected that data for the normal operation of a plant is available for optimizing the fault-free model of the plant. However, data for faulty behaviour are usually hard to obtain, particularly for faults that cause damage to the plant. For this reason it will often be necessary to use approximate plant fault models that are not optimized with respect to experiment. With this in mind, the models used to represent the three faults given in Section 4.2 were not optimized against experimental results obtained under faulty conditions, even though they were available in the present case.

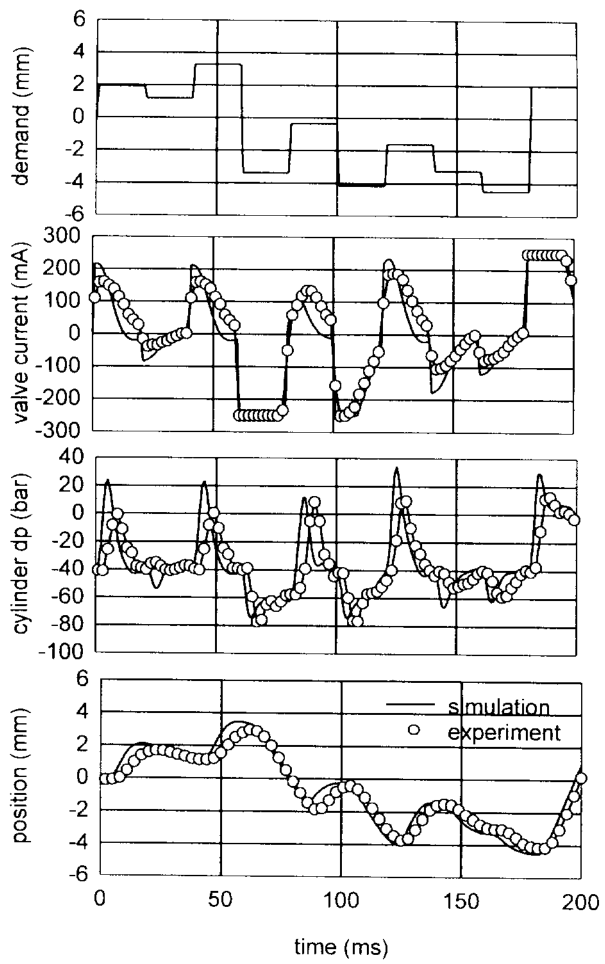


Fig. 6 Comparison of simulation and experimental data for the same demand signal

5 NEURAL NETWORK IMPLEMENTATION

5.1 Problem structure

When analysing the simple example given in Section 3, it was straightforward to obtain a full mathematical description of system behaviour and identify the system output variables required for diagnosis. With more complex real-life situations such as the actuator rig, the number of equations involved becomes large and it is hard to identify which system states should be measured. To facilitate this, a simplified schematic of the actuator system was developed (Fig. 7). The three faults of interest are captured in this model by the parameters P_s, k_l and c. Note that in this case it is not necessary to model the control loop since none of the fault parameters appear in the controller transfer function.

A mathematical description of this circuit is developed using the orifice equation for a single port of the control valve:

$$q = k_v x_v (P_s - P_p)^{1/2} \tag{4}$$

conservation of flow into the cylinder:

$$q - \dot{x}A_p - P_p k_l - \frac{V}{B_f}\dot{P}_p = 0 \tag{5}$$

and Newton's second law for mass and piston:

$$P_p A_p - \dot{x}c = m\ddot{x} \tag{6}$$

According to the above model, the parameters P_s, k_l and c, and hence the faults of interest, can be found if observations of the following variables are available: x_v, P_p, $\dot{P}_p$, $\dot{x}$, $\ddot{x}$. The time evolution of these variables forms a five-dimensional hypersurface in output vector space, the orientation of which depends on the values of the faulty parameters.

While it is possible to obtain these outputs from the simulation, the only similar output variables easily obtained from the rig are x, i_v (which carries information similar to x_v) and ΔP_p (which carries information equivalent to P_p). In a sampled data system derivatives are implicitly available from current and previous values (although due care must be taken with noisy signals).

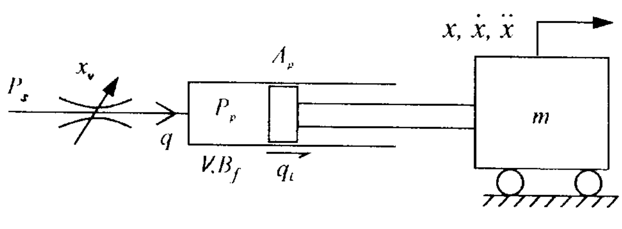


Fig. 7 Reduced actuator circuit model

Within an industrial environment it is important that a diagnosis system is robust with respect to measurement noise. To this end, it is useful to use neural network training data with the same noise characteristics as the actual plant data. In the present experiments, however, the signal-to-noise ratios for the rig data were high and it was appropriate to train the networks with noise-free simulation data.

5.2 Training data and network architecture

Visual inspection of the experimental data showed features of interest up to frequencies of around 100 Hz. To give adequate resolution of these features, a sampling rate of $10 \times 100 = 1000$ Hz was chosen. The sample size was chosen on the basis that each sample should be representative of the whole data. Visual inspection suggested that a 50 point sample (50 ms) would be a reasonable starting point. Trial and investigation after the fact showed these initial guesses to be acceptable.

The network architecture used for diagnosis is illustrated in Fig. 8. A sample of 50 values of x, i_v and ΔP_p forms a single training vector containing 150 values. This training vector is a specific example of the system output vector space for a given fault at a particular fault magnitude. To produce a training data set for a single-fault level, 200 training vectors at 5 ms increments were obtained. Each fault was divided into five different fault levels; thus a complete description of a fault required a total of 1000 training vectors.

The use of a 5 ms increment value for the data window was found to be an efficient use of 'limited' training data. This is more likely to be of value when training the net with experimental data, as described in Section 6.3.1. However, the size of the increment is not critical and other values may be used; its size does not appear to have any significant impact on training. The choice of 10 hidden neurons was based on trial and error. A larger number of hidden neurons led to potentially better accuracy at the expense of longer training times and the potential for overtraining (overfitting of the data). Fewer hidden neurons give shorter training times but less accuracy. Experience suggests that it is best to start with too few hidden neurons and then increase the number until accuracy improvements diminish.

Networks were trained using a standard error back-propagation algorithm with learning rate 0.2 and momentum 0.8. Weights were updated after all patterns in the training data had been presented.

6 RESULTS

6.1 Signal processing issues

In the following, neural network inputs were scaled between 0 and 1. Outputs are shown scaled back to the correct physical units. To improve clarity, graphs do not show every consecutive neural network output. No other signal processing is used.

6.2 Diagnosis of simulated actuator system

As an initial simplification, *individual* networks were trained for each fault. Figure 9 shows the results of querying a 150 : 10 : 1 network trained to recognize a leakage fault. The network has been queried with the original training data. Note that each second's worth of data represents a different (non-contiguous) data set

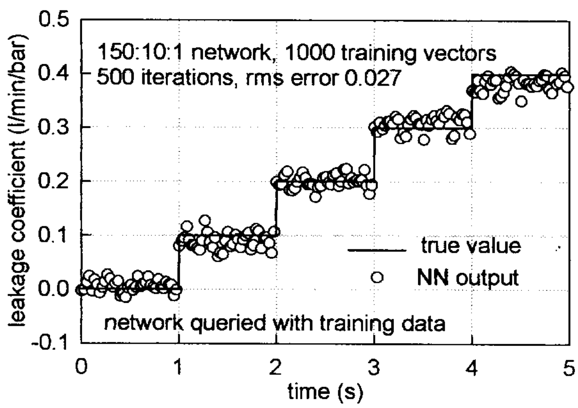


Fig. 9 Query results for the single leakage fault network: simulation data

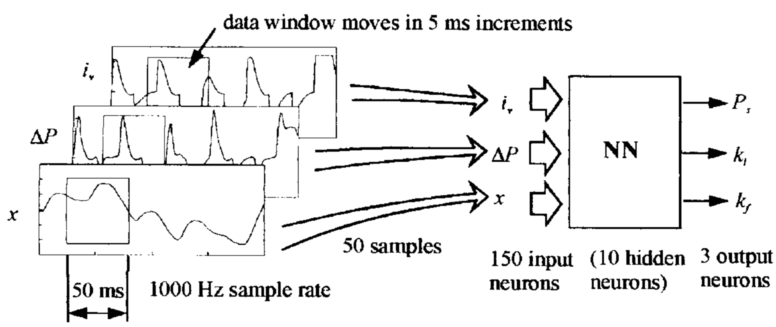


Fig. 8 Input/output architecture for the neural network

which are shown sequentially for comparative purposes only. Extensive investigations with single-fault networks were used to optimize both the size of the data window (and hence the number of input neurons) and the number of hidden neurons in the network.

The trained network was also validated for fault levels in between those for which it was trained. Results showed that networks were capable of generalizing for arbitrary fault magnitudes within the range of fault levels used for training.

When training separate nets for each fault it was found that some faults were much easier to learn than others. For example, for a net with 10 hidden neurons, to reach an r.m.s. error of 0.026 took 300 iterations for the supply pressure fault, 500 iterations for the leakage fault, but over 30 000 iterations for the friction fault. In each case, the symptoms caused by each fault were qualitatively similar in size and it was not clear why the friction fault took so much longer.

Individual networks solely trained for each fault, in this multiple-fault environment, were found to be unreliable when presented with data for faults other than those for which they were trained. One possible solution to this problem is to train individual networks using data for all the faults; i.e. each network must learn to accurately classify data relevant to its particular fault and to classify data from other faults as 'normal'. However, this approach requires all the individual networks to be trained with all the data. A simpler alternative is to employ a single network with an output neuron for each fault (as shown in Fig. 8), with this single network being trained on all the data. This approach has been adopted for the rest of the work reported in this paper.

Figure 10 shows results for querying a *single* network trained to recognize simulated supply, leakage and friction faults. The network has three output neurons and the scaled output of these is shown in separate graphs. Note that the results are concatenated from four different simulation runs and are therefore not contiguous. The net was trained for 200 iterations, with an elapsed time of 30 min.

An important result is that the output neuron for each fault indicates 'normal' when presented with both normal data and when presented with data corresponding to another fault. Thus, for a given fault, only the correct output neuron shows significant deviation from its normal value.

Since the net was only trained for a relatively small

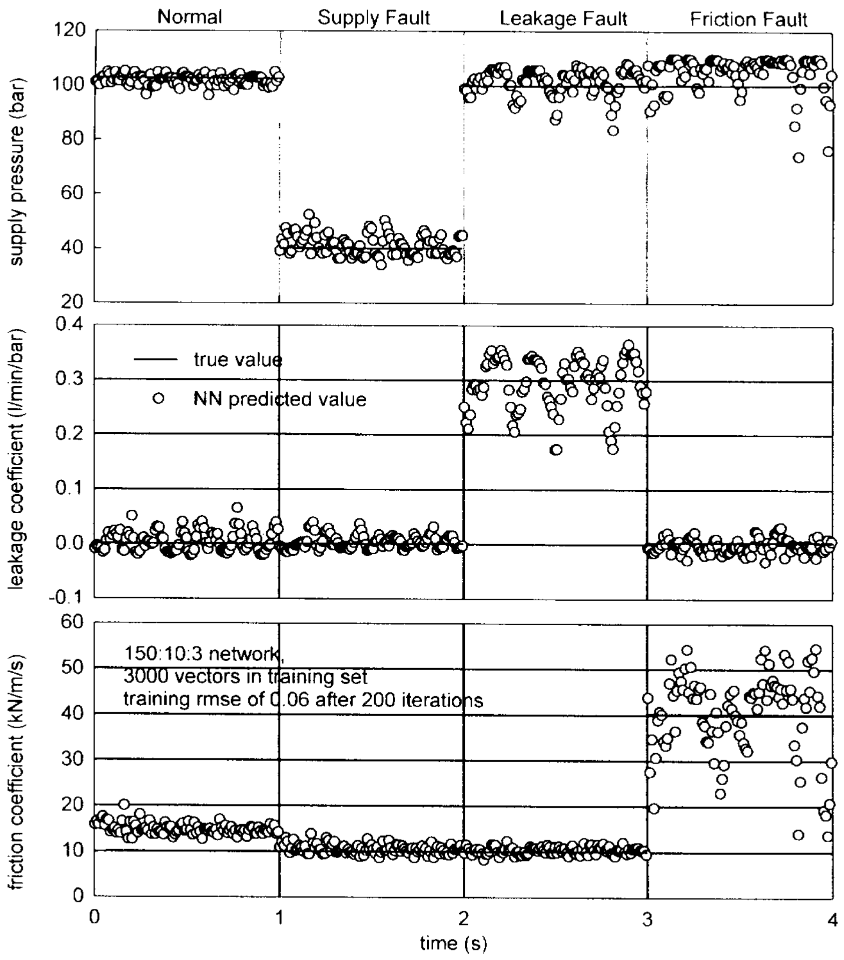


Fig. 10 NN trained on simulated data and validated with simulation data, single net for three faults

number of iterations, the results for the faults that are easiest to learn are better than the results for the faults that take longer to learn. In particular, the output for the friction fault shows a lot of scatter compared to the other two faults. Better results can be achieved by using more hidden neurons and training for longer. However, experience shows sharply decreasing returns in terms of the final r.m.s. error compared to the amount of time taken for training. For example, to halve the r.m.s. error to 0.03 requires approximately 100 times as much training time (2 days).

6.3 Diagnosis of experimental actuator system

6.3.1 Network trained with experimental data

Having optimized network structures using simulation data, similar networks were trained with experimental data. For comparison with Fig. 9, Fig. 11 shows output from a net trained to recognize a leakage fault in experimental data. Note that the y axis scale on this plot is an arbitrary and non-linear measure of the magnitude of the fault injected into the rig related to the fractional opening of the cross-line leakage restrictor. Both networks have the same structure and were trained for the same number of iterations. The final r.m.s. error for the net trained on experimental data is approximately twice that of the net trained on simulation data. The higher error in the experimental data is to be expected due to the presence of additional noise and unmodelled higher order dynamics.

The training times for the different faults using experimental data did not show the same trends as for simulation data. For the experimental data, the leakage fault proved easiest to learn and the friction fault could actually be learnt faster than with the simulation data. It is not clear why this is the case, but inaccuracies in the simulated fault models are a possible cause.

Figure 12 shows results for a single net trained to recognize all three faults in experimental data. Each separate plot shows data for 5 s of output from the rig. Faults were inserted by allowing the rig to run normally for approximately 2 s and then manually applying a rapid change to the setting of the supply relief valve or bleed restrictors. Note that during the insertion of the faults (fault transient) the neural network is still able to give a sensible output so that the prediction tracks the fault transient. This is possible because the data window of 50 ms used to make a diagnosis is relatively short compared to the time-scale of the transient. The results demonstrate the ability of the trained net to generalize by virtue of satisfactory ‘interpolation’ between the specific levels of fault used in training. It may be noticed that the supply fault output neuron of the net saturates when the net is queried with friction fault data. This implies that a friction fault shows symptoms similar to a high supply pressure fault. Increased hidden neurons and longer training alleviates, but does not completely remove, the problem. Nonetheless, Fig. 12 demonstrates a reasonably good immunity to misdiagnosis.

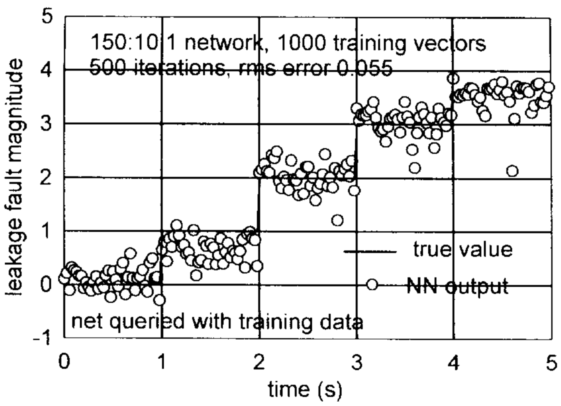


Fig. 11 Query results for the single leakage network: experimental data

6.3.2 Network trained with simulated data

The toughest test for a diagnostic network is for it to be trained with simulation data but queried with experimental data. To perform this test, the network in Fig. 10 was presented with the same data used to query the network in Fig. 12. The results of this are shown in Fig. 13. While the data show increased scatter, the simulation-trained network is clearly able to successfully diagnose the faults in the experimental data. Interestingly, of the three faults, the supply fault is the least-well diagnosed, which is surprising since it was anticipated that the simulation modelling of this fault was the most accurate.

Note that the y axis scales in Fig. 13 are given in terms of the parameter values used in the simulation. Thus, for example, the friction fault level is output in units of N s/m. It is not clear how reliable these scalings are since the net output depends on how the net generalizes on data it has not seen before.

7 SUMMARY OF APPROACH

Figure 14 shows a generalized flow-chart to aid development of diagnostic neural networks for systems similar to that studied in this paper. The first stage of the process (steps 1 to 3) is development of a validated model of the target system, including modelling of the faults of interest. This is standard engineering practice but can require considerable resources of knowledge and experience. Steps 4 and 5 are concerned with the problem of providing sufficient information for diagnosis in terms of both breadth and quality. The nature of this problem depends very much on the specific system and faults of interest and a successful solution may once again require

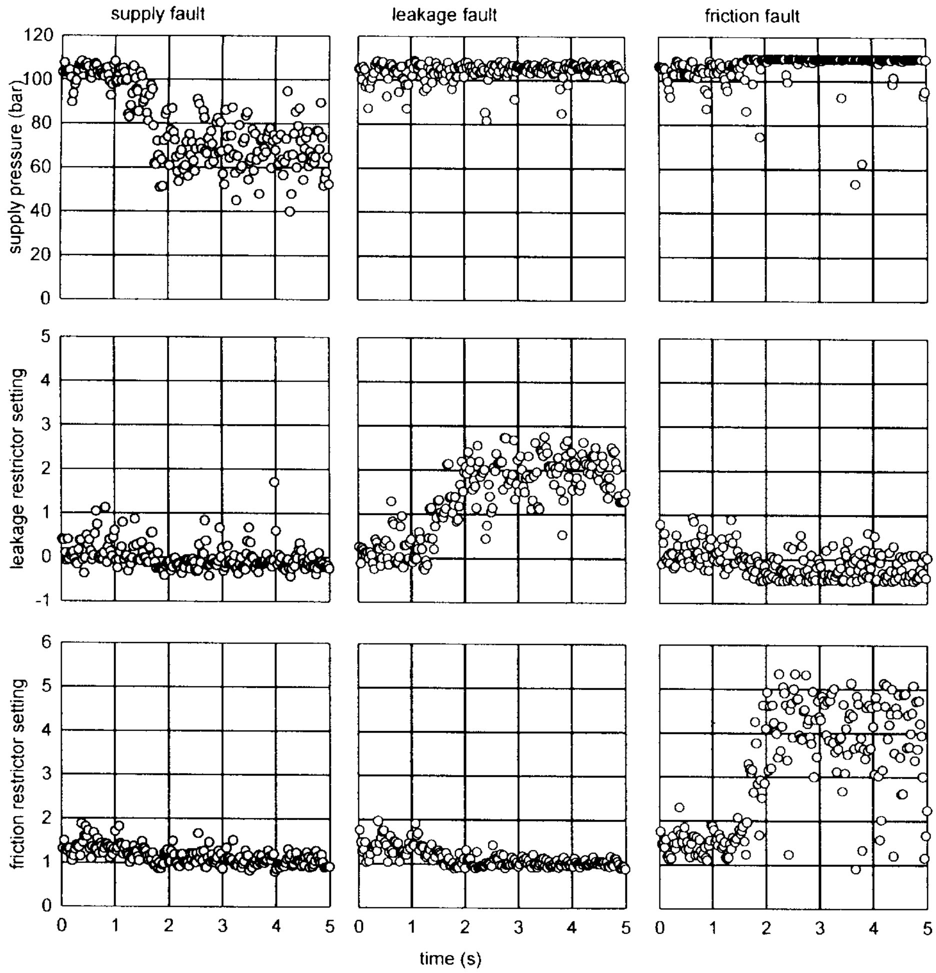


Fig. 12 NN trained with rig data and queried with rig data, single net for three faults: 150 : 10 : 3 network, 3000 training vectors, 200 iterations, r.m.s. error 0.07

considerable engineering judgement. The last steps in the process, that of training and validating a neural network, are relatively routine compared to the previous steps. Given a well-posed problem and a good data set, networks can be trained using any one of many standard software packages.

In practice it is hard to get each of the steps in Fig. 14 correct on the first attempt and some iteration is inevitable. The iterative loop in the modelling stage (steps 2 and 3) is relatively straightforward. However, if the neural network fails to converge at step 6, it can be for a number of reasons, and it may be necessary to return to various earlier stages in the process.

8 CONCLUSIONS

A simulation of a simple mass–spring–damper system was used to develop principles for classification-based diagnosis of dynamic systems using neural networks. These principles were applied to the diagnosis of a laboratory-based servo-valve-controlled, closed-loop hydraulic actuator system.

For the actuator system, neural networks were trained using both experimental data and data from a validated computer simulation. The fault-free simulation model of the actuator system was optimized with respect to experimental data using genetic algorithm techniques. Fault models for the simulation were developed without reference to experimental data.

Two approaches were employed for the training of networks for the diagnosis of single faults in a multiple-fault environment. Firstly, individual networks were trained for a specific fault and, secondly, a single network was trained with outputs representing all faults of interest. For each type of fault, the training data used covered five discrete fault magnitudes. Networks were not trained for simultaneous faults as this leads to a combinatorial explosion of training data and long training times. Network training showed similar trends for both simulation and experimental data, but for the same number of training iterations, higher final errors were exhibited in the experimental case.

Training separate nets for each fault alone allows quick and accurate training for single faults. Unfortunately, these nets do not give meaningful output when

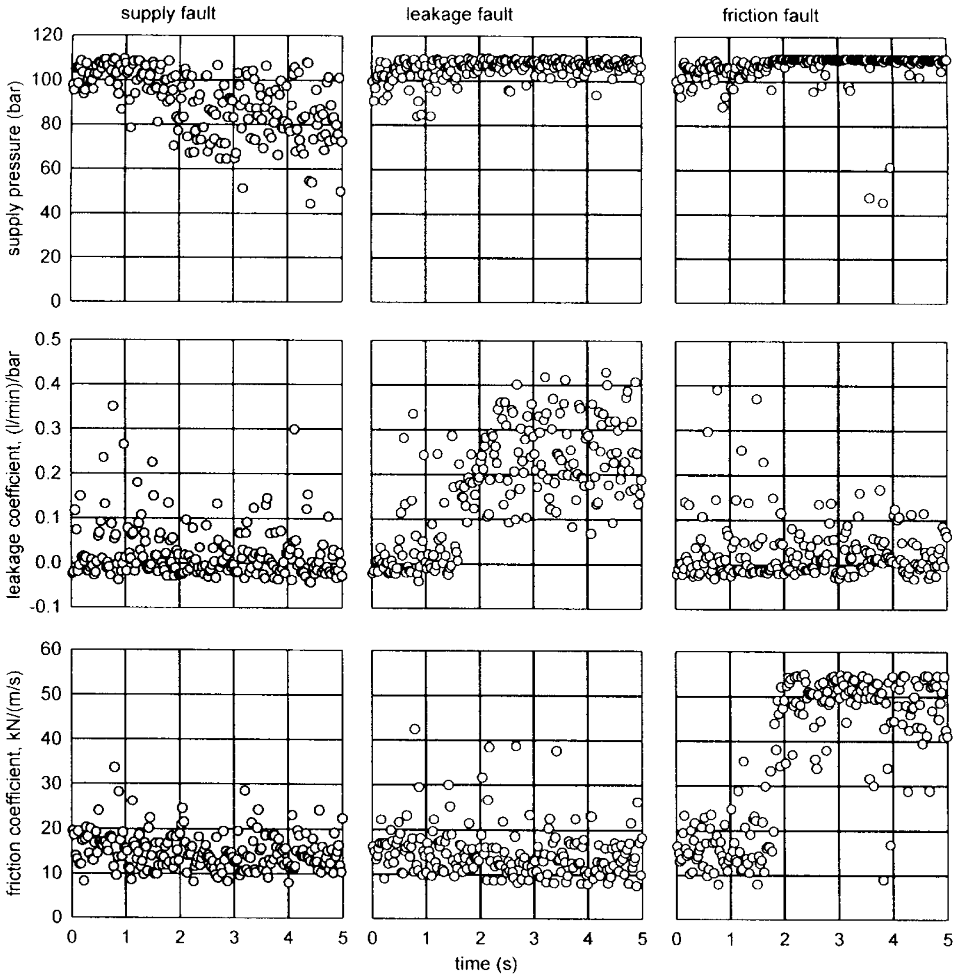


Fig. 13 NN trained with simulation data, queried with experimental data

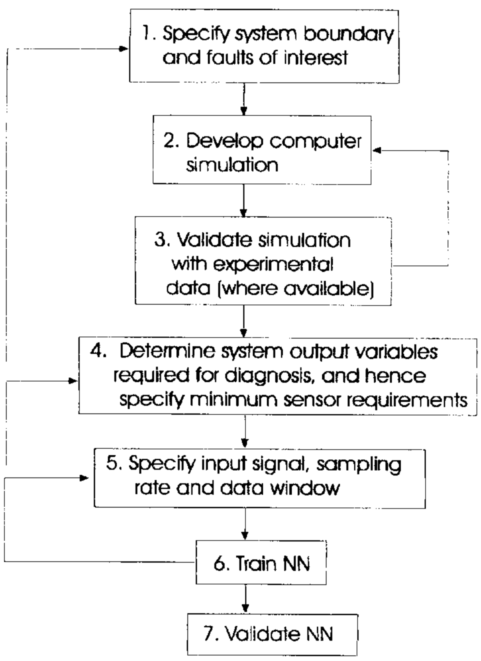


Fig. 14 Strategy for developing diagnostic neural networks for dynamic systems

queried with data corresponding to faults other than the fault for which it was specifically trained. However, three-fault networks correctly classify query data for all three faults. Specific-fault output neurons indicate normal when the net is queried with data for another fault. A three-fault network trained with simulation data was successfully used to diagnose faults in experimental data.

A major challenge is to obtain training data that cover the whole system output vector space for the faults of interest. This problem is directly related to that encountered in traditional system identification or parameter estimation schemes.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support for this project given by the Engineering and Physical Sciences Research Council through a research grant, reference GR/J58251. Thanks are also extended to Dr K. Pollmeier for optimizing the simulation models used in this research.

REFERENCES

1 **Fink, P.** and **Lusth, J.** Expert systems and diagnostic expertise in the mechanical and electrical domains. *IEEE Trans. Systems, Man and Cybernetics*, May 1987, **SMC-17**(3), 340–349.

2 **Wood, C. L.** FADES: a tool for automated fault analysis of complex systems. In *Research and Development in Expert Systems 89*, 1989, pp. 253–262 (Cambridge University Press).

3 **Watton, J., Lucca-Negro, O.** and **Stewart, J. C.** An on-line approach to fault diagnosis of fluid power cylinder drive systems. *Proc. Instn. Mech. Engrs, Part I, Journal of Systems and Control Engineering*, 1994, **208**(I4), 249–262.

4 **Far, B. H.** and **Nakamichi, M.** Qualitative fault diagnosis in systems with nonintermittent concurrent faults: a subjective approach. *IEEE Trans. Systems, Man and Cybernetics*, January 1993, **23**(1), 14–30.

5 **Linkens, D. A.** and **Wang, H.** Fault diagnosis on a qualitative bond graph model, with emphasis on fault localisation. In IEE Conference Publication 389, March 1994, pp. 1329–1334.

6 **Leitch, R. R.** Engineering diagnosis: matching problems to solutions. In Proceedings of International Conference on *Fault Diagnosis (Tooldiag '93)*, Vol. 3, Toulouse, France, 1993, pp. 837–844.

7 **Leitch, R. R., Chantler, M. J., Shen, Q.** and **Coghill, G. M.** A preliminary specification methodology for model based diagnosis. In *Annals of Mathematics and Artificial Intelligence*, Special Issue on Current Trends in Research on Model-Based Diagnosis, 1993.

8 **Doherty, N. F.** and **Kochhar, A. K.** Knowledge engineering for model-based diagnosis: an experience-based approach. *Engng Applic. of AI*, 1994, (6), 653–663.

9 **Doherty, N. F.** and **Kochhar, A. K.** MIDAS: an application of model-based reasoning for the diagnosis of hydraulic systems. *Knowledge-Based Systems*, June 1994, **7**(2), 127–134.

10 **Milne, R.** Strategies for diagnosis. *IEEE Trans. Systems, Man and Cybernetics*, May 1987, **SMC-17**(3), 333–339.

11 **Milne, R.** Artificial intelligence for online diagnosis. *IEE Proc.*, July 1987, **134**(D4), 238–244.

12 **Wilsky, A.** A survey of design methods for failure detection in dynamic systems. *Automatica*, 1976, **12**, 601–611.

13 **Patton, R., Frank, P.** and **Clark, R.** *Fault Diagnosis in Dynamic Systems; Theory and Applications*, 1989 (Prentice-Hall, Englewood Cliffs, New Jersey).

14 **Patton, R.** and **Chen, J.** Optimal unknown input distribution matrix selection in robust fault diagnosis. *Automatica*, 1993, **29**(4), 837–841.

15 **Isermann, R.** On the applicability of model-based fault detection for technical processes. *Control Engng Practice*, 1994, **2**(3), 439–450.

16 **Yu, D., Shields, D. N.** and **Mahtani, J. L.** A non-linear fault detection method for a hydraulic system. In IEE Conference Publication 389, March 1994, pp. 1318–1322.

17 **Bernieri, A., D'Apuzzo, M., Sansone, L.** and **Savastano, M.** A neural network approach for identification and fault diagnosis on dynamic systems. *IEEE Trans. Instrum. Measmt*, December 1994, **43**(6).

18 **Daley, S.** and **Wang, H.** On the application of neural networks to the monitoring of a simulated hydraulic rotary drive system. In 6th Bath International Fluid Power Workshop, University of Bath, 1993.

19 **Scaife, M. W., Charlton, S. J.** and **Mobley, C.** A neural network for fault recognition. In SAE International Congress and Exposition, Detroit, Michigan, March 1993, paper 930861.

20 **Leonard, J. A.** and **Kramer, M. A.** Radial basis function networks for classifying process faults. *IEEE Control Systems*, April 1991, 0272-1708/91, 31–38.

21 **Sorsa, T.** and **Koivo, H.** Application of artificial neural networks in process fault diagnosis. *Automatica*, 1993, **29**(4), 843–849.

22 **Patton, R. J., Chen, J.** and **Siew, T. M.** Fault diagnosis in non-linear dynamic systems via neural networks. In IEE Conference Publication 389, March 1994, pp. 1346–1351.

23 **Adam, S.** and **King, R.** A neural network-based architecture for on-line fault diagnosis. In *Parallel Distributed Computing in Engineering Systems* (Eds S. Tzafestas, P. Borne and L. Grandinetti), 1992 (Elsevier Science Publishers BV (North-Holland), Amsterdam).

24 **Becraft, W.** and **Lee, P.** An integrated neural network/expert system approach for fault diagnosis. *Computers in Chem. Engng*, 1993, **17**(10), 1001–1014.

25 **Bishop, C.** *Neural Networks for Pattern Recognition*, 1996 (Oxford University Press).

26 **Rumelhart, D. E., Widrow, B.** and **Lehr, M. A.** The basic ideas in neural networks. *Commun. ACM*, March 1994, **37**(3), 86–92.

27 **Minsky, M.** and **Papert, S.** *Perceptrons—An Introduction to Computational Geometry*, 1969 (MIT Press, London).

28 **Rumelhart, D., Hinton, G.** and **Williams, J.** Learning internal representations by error propagation. In *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, Vol. 1, *Foundations*, 1987 (MIT Press, London).

29 **Blum, E.** and **Li, L.** Approximation theory and feed-forward networks. *Neural Networks*, 1991, **4**(4), 511–515.

30 **Masters, T.** *Practical Neural Network Recipes in C++*, 1993 (Academic Press, New York).

31 **Crowther, W.** Fault diagnosis of engineering systems using neural networks: a practical approach. In IEE Colloquium on *Modelling and Simulation for Fault Diagnosis*, University of Leicester, September 1996, paper 1996/260.

32 **Ljung, L.** *System Identification Theory for the User*, 1987 (Prentice-Hall, Englewood Cliffs, New Jersey).

33 **Pollmeier, K.** Simulation and condition monitoring of fluid power systems. PhD thesis, University of Bath, 1997.

