

Sensor fault diagnosis for automotive engines with real data evaluation

M. S. Sangha¹, D. L. Yu^{*2} and J. B. Gomm²

¹Cummins Engine Co. Ltd, Royal Oak Way South, Daventry, NN11 8NU, UK

²Control Systems Research Group, School of Engineering, Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK

*Corresponding author: Email d.yu@ljmu.ac.uk

Abstract

In this paper, a new fault diagnosis method using an adaptive neural network for automotive engines is developed. A radial basis function (RBF) network is used as a fault classifier with its widths and weights on-line adapted to cope with model uncertainty and time varying dynamics caused by mechanical wear of engine parts, environment change, etc. Five different sensors are investigated for an automotive engine including throttle angle, manifold pressure, manifold temperature, crankshaft speed and engine torque. The engine data is acquired from a one-litre Volkswagen petrol engine test bed under different operating states, and then simulated multiplicative faults are superimposed. The real data experiments confirm that sensor faults as small as 2% can be detected and isolated clearly. The developed scheme is capable of diagnosing faults in on-line mode and can be directly implemented in an on-board engine diagnosis system.

Keywords: Sensor faults, Fault diagnosis, Adaptive neural networks, Fault classification, IC Engines, Radial Basis Function, On-board FDI.

DOI: <http://dx.doi.org/10.4314/ijest.v3i8.2>

1. Introduction

Advanced methods of monitoring, fault detection and diagnosis are becoming increasingly important for the reliability, safety and efficiency of automotive engines. There are strict rules and regulations, to be abided by all car manufacturers, for low emissions to limit air pollution. A small bias in a sensor can adversely affect the engine efficiency especially when the sensed value is directly or indirectly utilised in the electronic control unit (ECU) of the engine for engine control. Some sensor faults can even lead to increased pollution due to less efficient engine operating state. Timely detection of faults can prevent the development of possible catastrophes in the near future. Some sensor faults that are used for feedback control will affect both dynamic and steady state performance. Therefore, sensor fault detection and isolation is important for automotives.

There are a number of fault diagnosis systems in practice but major car firms are now looking at neural networks to solve the demanding engine control and diagnostic requirements (Evans-Pughe, 2006). For instance Ford has introduced the Econoline van that uses a neural net to detect misfire in its V10 engine. Applications of artificial neural networks (ANNs) to engine modelling and control have previously been presented, e.g. (Tan and Saif, 2000; Kimmich et al., 2005; Manzie et al., 2001; Jakubek and Strasser, 2002). Earlier work on fault diagnosis of an automotive engine based on parity equations derived from an engine model was presented in (Gertler et al., 1993). The application of data-driven monitoring techniques to accurately diagnose air leakage in the inlet manifold plenum chamber of an automotive engine with a diameter size as small as 2 mm can be found in (Antory, 2007). A continuous wavelet transforms technique for the fault signal diagnosis in an internal combustion (IC) engine and its cooling system was presented in (Wu and Chen, 2006.). A neural network model-based fault classification system for a nonlinear dynamic process was investigated in (Yu and Gomm, 1999; Yu et al., 2005) and the real data experiment showed that sensor faults could be detected and isolated even without a process mathematical model. A fault detection and isolation (FDI) scheme for abrupt and incipient faults using online estimators (Zhang et al., 2002) is another good example of automated fault-diagnosis methodology.

Many authors have previously investigated fault diagnosis of sensor faults (Vemuri 1999; Muldoon et al., 2002; Ding et al., 2004; Crossman et al., 2003) in different ways. A robust sensor fault diagnosis algorithm for a class of nonlinear dynamic system

using adaptive techniques to establish the unknown constant sensor bias in the presence of system modelling uncertainties and sensor noise was presented in (Vemuri 1999). The robustness of the algorithm is demonstrated by illustrating a simulation example of universal exhaust gas oxygen sensor bias. In Muldoon et al. (2002), a virtual fusion/estimation technique was proposed, which provided basic signal redundancy and fault tolerance. The dynamic vehicle sensor information was used to develop a fusion-processing algorithm in Matlab. In Ding et al. (2004) a model based sensor monitoring scheme for the Electronic Stability Program (ESP) system was presented which was developed and produced in series with the company, Continental Teves. Analysis of vehicle faults relating to signal behaviour, signal segmentation, feature extraction and selection with a focus on the power-train control module (PCM) signal fault diagnosis was presented in (Crossman et al., 2003).

In this paper five different sensors in automotive engine have been investigated for positive and negative bias faults. The sensors considered are throttle angle position, crankshaft speed, torque, inlet manifold pressure and temperature sensors. The faults considered are realistic and have been considered previously (Antory, 2005; Nyberg and Stutte, 2004; Capriglione et al., 2002; Capriglione et al., 2003; Capriglione et al., 2004; Vinsonneau et al., 2001). The throttle angle position, crankshaft speed and manifold pressure sensors have also been previously investigated in Capriglione (2003) and implemented on a digital signal processor (DSP). The applicability to on-board control units for fault diagnosis using fixed artificial neural networks is demonstrated. However, fixed-parameter neural classifiers cannot deal with engine-to-engine variations, parameter uncertainty, disturbances, variations due to environmental changes and aging of the engine and measurement devices. In this paper an adaptive on-line fault diagnosis strategy is evaluated on real engine data to cope with these effects.

A one-litre Volkswagen car engine test bed is used for the entire experimentation in this study. The relevant technical specifications of the engine test bed are tabulated in Section 2. Several sets of real engine data are acquired from the engine test bed in different operational modes in Section 3. The data is acquired on different days to ensure that the ambient conditions are different each time. A fault diagnosis methodology via adaptive neural nets is described in Section 4. The initial development of the method with computer simulations was given in (Sangha et al., 2006). It is impracticable for the authors to get real faulty data from a running engine at specific times and situations for as small as 2% variations. Therefore sensor faults are realistically simulated (Section 5.1). The adaptive neural net is initially trained on one of the real data sets for no fault and ten sensor faults (Section 5.2). The trained neural network is then tested for all the sensor faults for $\pm 2\%$ bias on different engine data sets and satisfactory diagnosis results are achieved (Section 5.3). It is important to mention here that sensor failures investigated in (Capriglione et al., 2007), e.g. short circuit (zero deflection), open circuit (full deflection), hold (reading-halt) and short circuit between two sensors, are not considered in this research due to the fact that these are easier to be diagnosed compared to a sensor bias fault as small as 2%. The adaptive neural classification system has also been tested for $\pm 1\%$ sensor bias faults. But, misclassifications appear due to the too small fault signal compared to the uncertainty. Finally the conclusions are drawn in Section 6.

2. Experimental Set-up and Engine Specifications

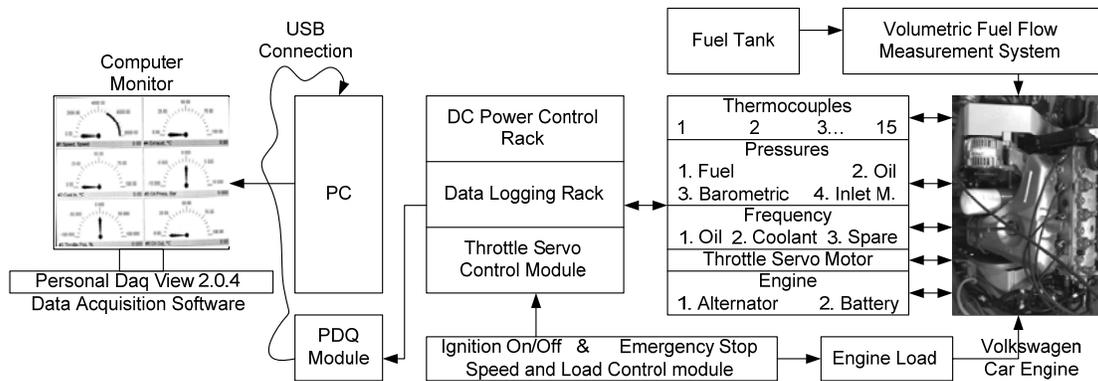


Fig. 1 Schematic diagram of engine test bed

A Volkswagen petrol car engine test bed is used for real engine data collection. Fig. 1 shows the schematic diagram of the experimental set-up. Main details of the engine specifications are given in Table 1. The engine test bed has a provision for 15 thermocouple inputs, 24 digital I/O and 2 frequency/pulse differential signals from transducers at different locations in the entire engine system.

Table 1: Volkswagen car engine specifications

Engine Code	Number of Cylinders	Nominal output in kW/rpm	Maximum torque N-m/rpm	Bore/Stroke in mm	Capacity in cm ³	Compression
ATE	4	37/5000	86/3000... 3600	67.1/70.6	999	10.5

The interface hardware of the test bed, Personal Daq (PDQ), provides a connection between the computer USB port and the data logging rack. The data logging rack is directly connected to the transducer connection panel through cables. All the transducers installed in the engine test bed are connected to the transducer panel to provide an interface with the data logging rack.

The data acquisition software, Personal Daq View, is configured on a PC to access real engine data through a USB port. This software has a configurable user-friendly interface which can provide a highest frequency/pulse measurement up to 1 MHz. Up to 100 Personal Daq/PDQ (combined units) can be connected to one PC by the use of USB hubs; providing a total channel capacity of 8,000 channels. With slow measurement duration of 610 milliseconds, there will only be 1.6 samples per second whereas with a very fast measurement duration of 12.5 milliseconds, there can be 80 samples per second at the most. The sampling time is also dependent on the number of different variables to be acquired. A sample time of 256 ms was found appropriate for this experiment to catch transient dynamics of the variables required for fault diagnosis.

3. Table Data Acquisition

For fault detection and isolation via neural networks, training and testing data sets are required and all of which should contain samples with no fault and all faults. It is impracticable to produce real faults during engine operation and to acquire training data for every possible operating mode of an engine. Therefore a suitable set of possible faults was simulated on the basis of data acquired in fault-free condition which is explained in section 5.1.

Data acquisition is one of the most important parts of these experiments. The data is acquired in many different operating states of the engine to cover the entire range of normal engine operation. The engine is carefully operated to run in different speed and torque conditions to cover the maximum spectrum of real engine runs on road. The different sets of acquired engine data are shown in Table 2.

Table 2: Ranges of important variables in acquired real engine data

	Variation in % throttle angle position	Variation in crankshaft speed (RPM)	Variation in torque (N-m)
Data sets 1 & 2 (High speed variation run)	0.28 ~ 77	720 ~3976 (High variation)	-1.0 ~ 85
Data sets 3 & 4 (Approximately constant speed run)	20 ~ 50	2127 ~ 2192 (Almost Constant)	28 ~ 48
Data sets 5 & 6 (Low speed variation run)	4 ~ 80	1510 ~ 2870 (Low variation)	-0.25 ~ 85.8

The speed variation during experiments covers a wide range from nearly 700 rpm to 4000 rpm. The torque variation ranged from -1.0 N-m to 86 N-m. The maximum allowable torque for this engine is 86 N-m; and therefore the engine was run up to the maximum allowable torque. The engine data was acquired on different days to ensure different ambient conditions. Six different data sets were collected on different days and different sessions for training and testing of neural classifier. One raw data set is graphically shown in Fig 2 for illustration.

The sampling time for the raw data in Fig 2 was 1.4 seconds and a total of 441 samples were acquired and recorded on computer using the Personal Daq View software. Initially the engine was run idle for 10 minutes for proper heating up and then throttle position and load were simultaneously changed and necessary readings were recorded for 617.4 seconds as shown graphically in Fig. 2. In the beginning, the engine is run on low throttle (slow speed) for a minute as if run in congestion and then, gradually, the throttle is increased to increase speed up to nearly 4000 rpm to represent highway run. Then, the throttle is gradually decreased in steps to reduce the engine speed and this is how a high variation in engine speed is achieved. The load on the engine is also

changed within allowable limits during data recording to achieve up hill and down hill running conditions of the engine. An increase and decrease in engine load corresponds to up hill and down hill runs of the engine respectively.

Some engine operating modes, e.g. sudden and harsh braking, steep up hill and steep down hill runs, have not been considered in these experiments due to limitations of the technical facilities available in the workshop. The ranges for all the raw data are tabulated in Table 2.

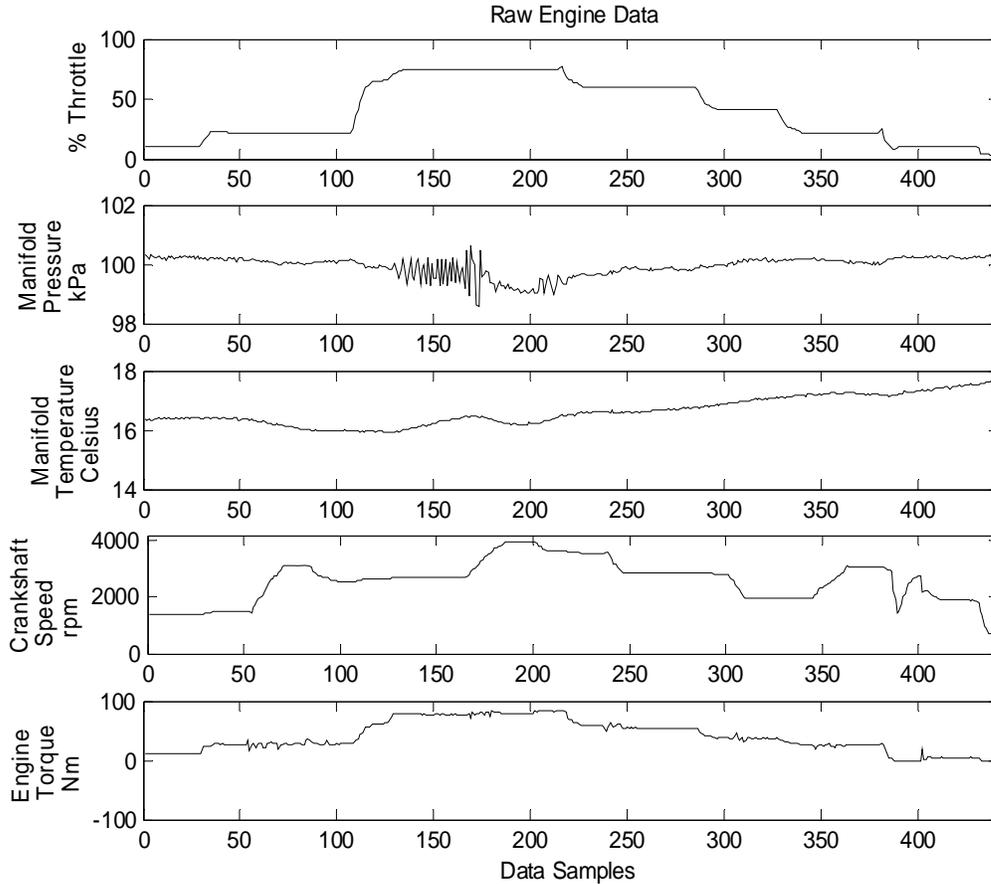


Fig. 2 Raw engine data in data set 2

4. Fault Diagnosis Method

An RBF network with its widths and weights on-line adapted is used in this research to classify sensor faults. For completion the RBF network is briefly reviewed here. The structure of RBF network is shown in Fig. 3 and it consists of three layers: input, hidden and output. The input layer simply receives the network input vector $x \in \mathcal{R}^d$, and passes the inputs to each node in the hidden layer. The hidden layer consists of n_h nodes that process the input vector. The i^{th} node in the hidden layer contains an individual centre vector c_i of the same dimension as x and a scalar width ρ_i . The Euclidean distance between the input and the centre vectors is calculated,

$$z_i = \|x - c_i\| = \sqrt{(x_1 - c_{i1})^2 + \dots + (x_n - c_{in})^2}$$

where $i = 1, \dots, n_h$, and passed through a non-linear basis function to produce the hidden node outputs ϕ_i . Several choices of basis function are available, e.g. thin plate spline, Gaussian function, etc. Gaussian basis functions provide a local excitation of the node with an output ϕ_i near zero for inputs far from the centre and ϕ_i near one for inputs close to the centre. This is especially suitable for classification applications and is therefore used in this work. The Gaussian basis function is defined as

$$\phi_i = \exp\left[-(z_i / \rho_i)^2\right] \quad \rho_i > 0$$

Finally the network outputs are computed as a linear weighted sum of the hidden node outputs:

$$\hat{y} = W\phi$$

where $\hat{y} \in \mathfrak{R}^q$ is the output vector, $W \in \mathfrak{R}^{q \times n_h}$ is the output layer weight matrix with element w_{ij} connecting the j^{th} hidden node to the i^{th} output, and ϕ is a vector containing the hidden node outputs.

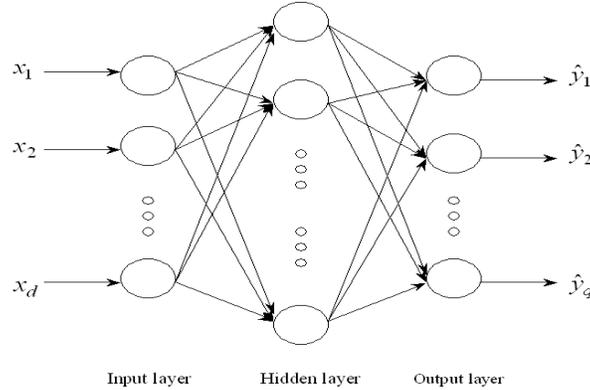


Fig. 3 The RBF neural network structure

The RBF network, as the fault classifier, receives all possible and relevant signals containing fault information. From engine dynamics it is known that all the five sensor outputs are involved in the dynamics and the interactions among them should not be neglected. Therefore, all the five sensor variables, the throttle angle position, the manifold pressure, the manifold temperature, the crankshaft speed and torque, are chosen as neural network (NN) inputs. Also, when the dynamics of the engine are considered, the delayed sensor outputs should be included, where the number of delayed samples for each variable should be chosen according to the order of the dynamics. However, including the delayed outputs will greatly increase the number of network inputs and lead to a much larger network size, which requires more computing time and possible lower generalization ability. Here we use only current outputs as the inputs of the network, while leaving the dynamics issue to be solved by adaptation of the network in each sample period. This is possible due to the different patterns at different time instants caused by the system dynamics can be classified into the same category, with the network parameters adapted to proper values. The network has 11 outputs with each indicating one of the investigated states: one for no-fault state, 5 for positive bias of the 5 sensors and 5 for negative bias of the 5 sensors. The information flow for the fault diagnosis method is illustrated in Fig. 4.

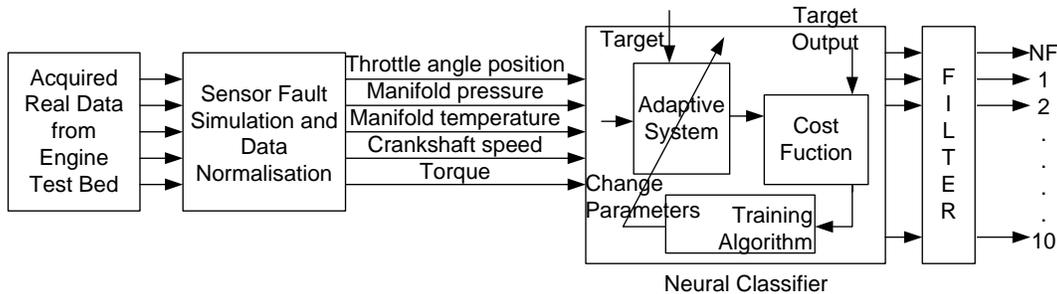


Fig. 4 Information flow of the fault diagnosis method

The acquired raw data is linearly normalised to the range of [0 1] and the different 10 sensor faults are simulated by superimposing on the normal sensor output with one fault at a time. Then all five inputs are fed to the adaptive classifier. Widths in the hidden nodes and the weights in the output layer of the RBF network are on-line adapted during the monitoring to minimise the sum-squared error between the output from the adaptive network and the pre-decided target output. Gradient descent method is used for adjusting the widths of the RBF network. The width in each hidden layer node is initially chosen as a constant using the P-nearest rule. The classification is sensitive to the Gaussian local function, which is mainly characterised by the width. Therefore, a gradient descent algorithm is derived to on-line adapt the widths to achieve a minimal objective function given as follows:

$$J = \sum_{j=1}^q e_j^2 \tag{1}$$

where $e_j = y_j - \hat{y}_j$ is the j^{th} classifier output error; and y_j is the j^{th} training target. The new updated value of the width can be achieved by the following equation Sangha et al. (2006):

$$\rho_i(k+1) = \rho_i(k) + 4\alpha\phi_i(k) \frac{\|x(k) - c_i\|^2}{\rho_i^3(k)} \sum_{j=1}^q e_j(k)w_{ij}(k) \tag{2}$$

where $x(k)$ is the network input vector at iteration k , c_i the centre of the i^{th} activation function, $\phi_i(k)$ is the Gaussian basis activation function, α is a learning factor and $0 < \alpha < 1$.

If the RBF network has d inputs, q outputs and n_h hidden nodes, the output matrix with N samples ($\hat{Y}^{N \times q}$) can be written as $\hat{Y} = \Phi(X)W$

where $X^{N \times d}$ is the input matrix, $\Phi(X)^{N \times n_h}$ is the matrix of activation function outputs and $W^{n_h \times q}$ is the matrix of weights. The recursive least squares (RLS) algorithms used for on-line training of the weights W . The RLS algorithm given in Ljung (1999) is summarized as follows.

$$L(k) = \frac{P(k-1)\varphi(k)}{\lambda(k) + \varphi^T(k)P(k-1)\varphi(k)}$$

$$W(k) = W(k-1) + L(k)[y^T(k) - \varphi^T(k)W(k-1)]$$

$$P(k) = \frac{1}{\lambda(k)} \left[P(k-1) - \frac{P(k-1)\varphi(k)\varphi^T(k)P(k-1)}{\lambda(k) + \varphi^T(k)P(k-1)\varphi(k)} \right]$$

where $y(k)$ and $\varphi(k)$ represent the network target vector and hidden layer output vector at sample instant k respectively. Also, $y^T(k)$ and $\varphi^T(k)$ are the k^{th} row vectors in matrices Y and Φ . $P(k)$ is the covariance matrix and $L(k)$ is the gain matrix, $\lambda(k)$ is called the forgetting factor and lies in the range of (0,1). The parameters $L(k)$, $W(k)$ and $P(k)$ are updated orderly with the activation function outputs $\varphi(k)$ at each sampling time after setting the value of $\lambda(k)$ and the initial values of W and P .

While the fault classifier diagnoses faults on-board, the classifier is adapted on-line so that the engine parameter uncertainty and especially the time varying dynamics caused by mechanical wear of components and environment change can be taken care of. In this way, the classification error and consequently the false alarms will be greatly reduced. Here, a false alarm is the detection of a fault caused due to noise, parameter uncertainty or time varying dynamics when actually there is no fault. The on-line adapted classifier is developed to cope with such situations, which were not considered by a fixed parameter classifier investigated by previous authors in Capriglione (2007).

The fault classification and on-line adaptation are implemented as follows. Firstly, the measurements are read into the electronic control unit (ECU). Then, the data is fed into the classifier to diagnose faults. After this, the target will be modified according to whether a fault or several faults are detected. If a fault is detected, the on-line training target vector will be changed to the target vector corresponding to the occurred fault. Then, the measurements and the modified target are used to update the classifier. In the adaptation, the width in each hidden node is adapted using the gradient descent algorithm in (2) and the centre locations remain fixed. This is followed by adaptation of the weights using the recursive least squares (RLS) algorithm.

To reduce the effect of peak noise on the fault detection so as to further reduce the false alarms, the mean absolute modelling error for each classifier output is calculated for the previous M samples as the residual,

$$r_j = \frac{1}{M} \sum_{i=k-M+1}^k |y_j(i) - \hat{y}_j(i)|, \quad j = 1, \dots, q \tag{3}$$

and a fault is believed to be fired when

$$r_j \geq r_t \tag{4}$$

where k is the sample instant, r_j is the residual and r_t is a threshold to be designed according to the noise level. Another point is that a multi-epoch training of the width in one sample period using the gradient descent method is employed.

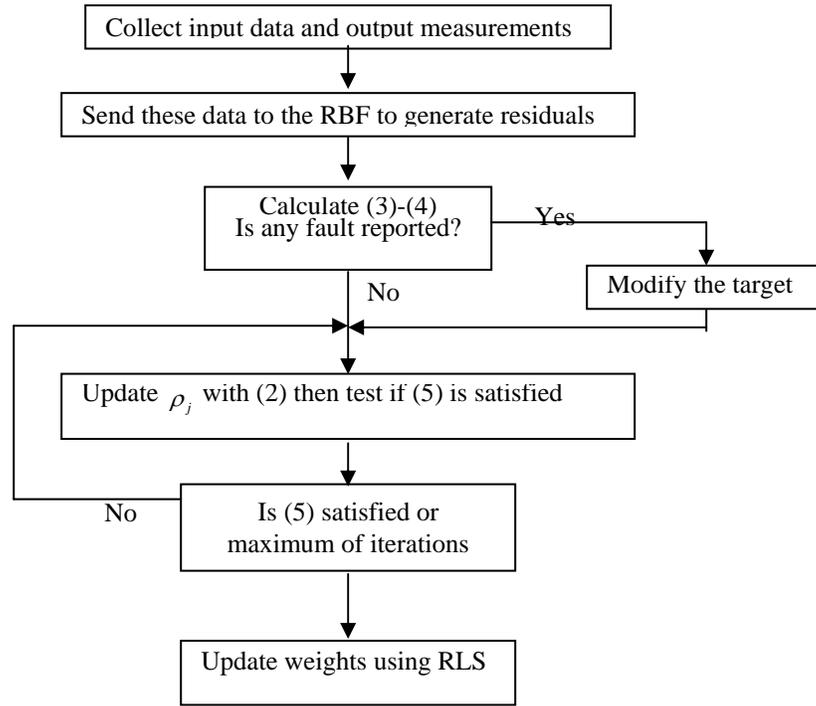


Fig. 5 Flow chart of fault diagnosis and classifier updating

It was found that a single iteration updating of the widths with the gradient descent method would not reach the minimum if the learning rate is chosen small, while a large learning rate may cause unstable convergence. The recursive updating of the widths therefore runs until the following is satisfied,

$$\left| \frac{\partial J}{\partial \rho_i} \right| \leq \sigma, \quad i = 1, \dots, n_h \tag{5}$$

where σ is a pre-specified small positive constant, or a pre-specified number of iterations is reached. The fault diagnosis and classifier adaptation within one sample period is illustrated in Fig. 5.

5. Fault Diagnosis

5.1 Fault Simulation on the Real Data

As real sensor faults with the amplitude as small as 2% and at some specific time periods are not easy to be introduced to the engine test bed in practice. Therefore, the sensor faults are simulated by superimposing a bias with a certain percentage of its normal sensor output on real data. This does not affect system dynamics and fault detection. Both +-2% and +- 5% bias are used and this is achieved using a multiplying factor (MF) of 1.02 and 0.98 and 1.05, 0.95 respectively. The 2% faults are shown in the Table 3 together with all the names of faults simulated.

Table 3: Simulated faults and their multiplying factors

State No.	Fault description	MF
1	No Fault (All the sensors reading correctly)	1.00
2	Throttle angle sensor 2% over reading	1.02
3	Throttle angle sensor 2% under reading	0.98
4	Manifold pressure sensor 2% over reading	1.02
5	Manifold pressure sensor 2% under reading	0.98
6	Manifold temperature sensor 2% over reading	1.02
7	Manifold temperature sensor 2% under reading	0.98
8	Crankshaft speed sensor 2% over reading	1.02
9	Crankshaft speed sensor 2% under reading	0.98
10	Torque sensor 2% over reading	1.02
11	Torque sensor 2% under reading	0.98

317 samples are kept in each data set with some samples at the beginning and end of the set. Then, one of 11 fault states is simulated to all the samples of the set. This is repeated for 11 times, to form $317 \times 11 = 3487$ samples in each faulty data set. In this way the fault classification can be tested and viewed clearly.

5.2 Network Structure Selection and Initial Training

Before the network is used on-line, it needs to be pre-trained off-line. This is referred to as initial training. With 5 inputs and 11 outputs determined in Section 4, the remaining structure of the network to be determined is the number of hidden layer nodes. Different numbers of hidden nodes ranging from 5 to 35 were tried and network performance, in terms of good classification and time taken for computations, was analysed. Twenty hidden nodes were found appropriate for a satisfactory level of performance because more hidden nodes gave a slightly better performance but the computational load on the microprocessor increases substantially in terms of real time processing. Therefore the structure of the network was taken as $5 \times 20 \times 11$.

The target of network output in the training is given by a 11-dimension row vector with 0 or 1, a 0 (zero) output implies that the fault does not occur while a 1 (one) output implies that the fault occurs. As 11 fault states are simulated with each for 317 data samples, one target vector will be used repeatedly for 317 times and the vector has only one "1" corresponding to the fault and all the other entries are "zero". Thus, target matrix X_o has $317 \times 11 = 3487$ rows and 11 columns. Each column of X_o represents i th target for one of the 11 states. Its first column has ones from the first row to the 317th row and the other entries are zeros, the second column has ones from the 318th row to the 634th row and the other entries are zeros, the last column has ones from the 3171st row to the 3487th row and the other entries are zeros. This is shown as follows:

$$\begin{array}{cc}
 \text{Row Numbers} & X_o \\
 1 \sim 317 & \left[\begin{array}{cccccc} 1 & 0 & \dots & \dots & 0 & 0 \end{array} \right] \\
 318 \sim 634 & \left[\begin{array}{cccccc} 0 & 1 & \dots & & & 0 \end{array} \right] \\
 \vdots & \left[\begin{array}{cccccc} \vdots & \vdots & \ddots & & & \vdots \end{array} \right] \\
 \vdots & \left[\begin{array}{cccccc} \vdots & & & \ddots & & \vdots \end{array} \right] \\
 3171 \sim 3487 & \left[\begin{array}{cccccc} 0 & 0 & \dots & & 0 & 1 \end{array} \right]
 \end{array}$$

Thus, the transpose of the i^{th} row in X_o is used as the training target vector for the i^{th} training pattern. The centers are chosen using the K -means clustering algorithm from the training data set. The widths were initially chosen using the P -nearest neighbour's algorithm, and the weights were trained using the RLS algorithm.

5.3 Fault Classification Results

The adaptive network is used to diagnose faults with test data sets after initial training with the training data set. The fault detection threshold in (4) was chosen as $r_t = 0.5$ with M in (3) chosen as $M = 3$. This replies that one of the network outputs must be over 0.5 continuously for 3 samples, a fault is then believed occurred. High thresholds may lead to missed detections whilst low thresholds may cause more false alarms. A proper choice can be found in experiment and $r_t = 0.5$ is found as a good compromise between reliability of detection and insensitivity to noise in this application. The threshold for the gradient of the objective

function in (5) was chosen as $\sigma = 0.00001$. The forgetting factor for the RLS algorithm was chosen as a constant value of $\lambda = 0.99$.

The adaptive network is initially trained on data set 6 for 5% sensor faults and tested on other data sets. The results for network testing on data set 1 with 2% fault are shown in Fig. 6.

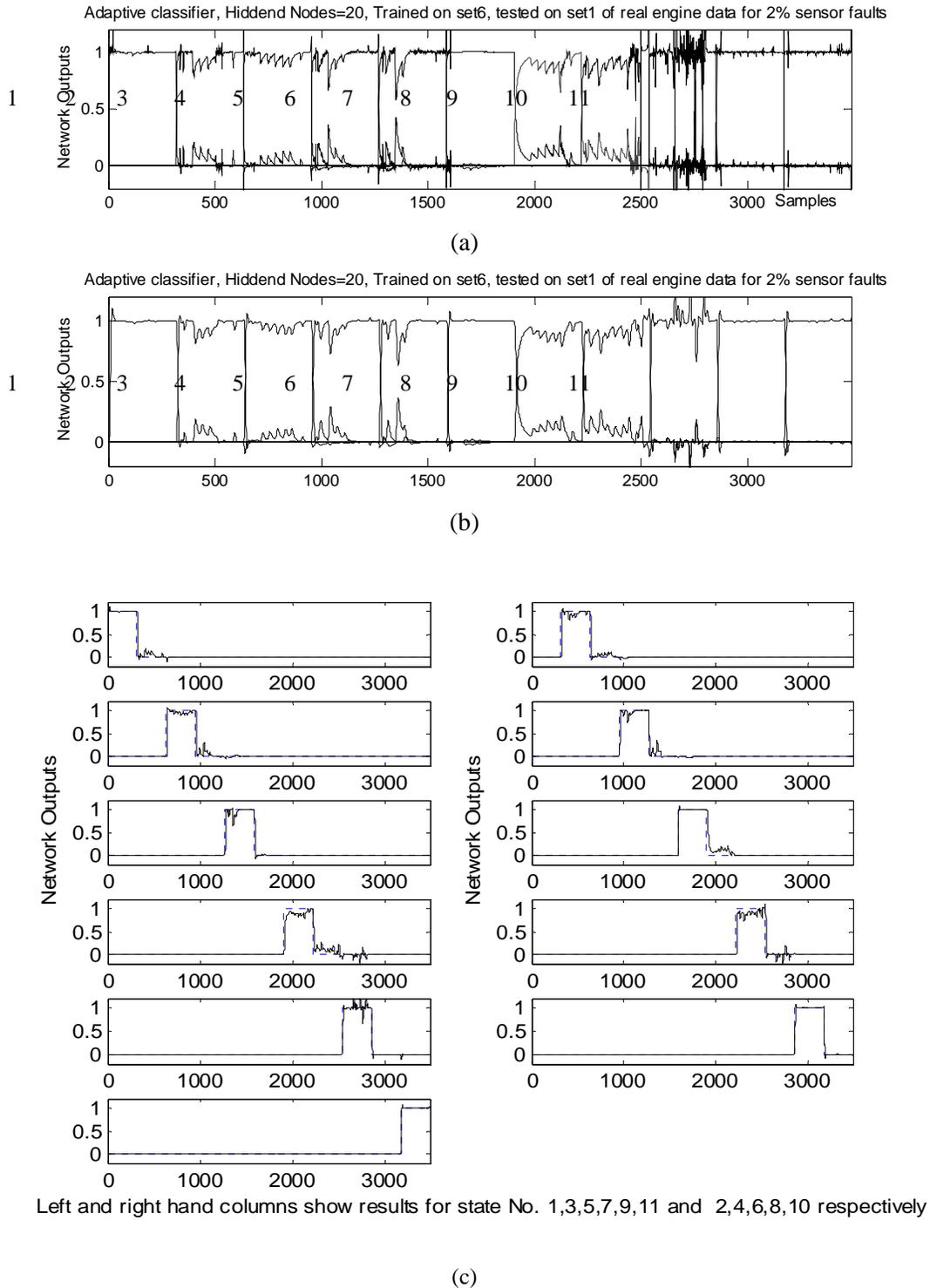


Fig. 6 Classification result for 2% faults when the network is trained on data set 6 and tested on data set 1. (a) Without data filtration (b) after low pass data filtration (c) Each state separately shown for clarity

There would be misclassifications for unfiltered data as shown in Fig. 6(a). There are a number of spikes crossing the threshold of 0.5 which may cause false alarms if their average value in three samples is over 0.5. To further reduce the false alarms the classifier outputs are filtered by a low-pass filter. The low-pass Butterworth filter attenuates the high frequency spikes and the result for the same fault is shown in Fig. 6(b). There are still a few spikes visible but these do not cross the threshold of 0.5 and therefore cannot cause a false alarm. To analyse the resultant classification, all the 11 states are shown separately in Fig. 6(c) for better visibility. It can be seen that all the ten fault states and the no fault state are clearly classified.

Another test result for network initial training on data set 6 with 5% faults and testing on data set 5 with 2% faults is shown in Fig. 7. Fig. 7(c) shows the results are satisfactory with all the ten sensor faults and no fault state clearly classified with no false alarms or misclassification.

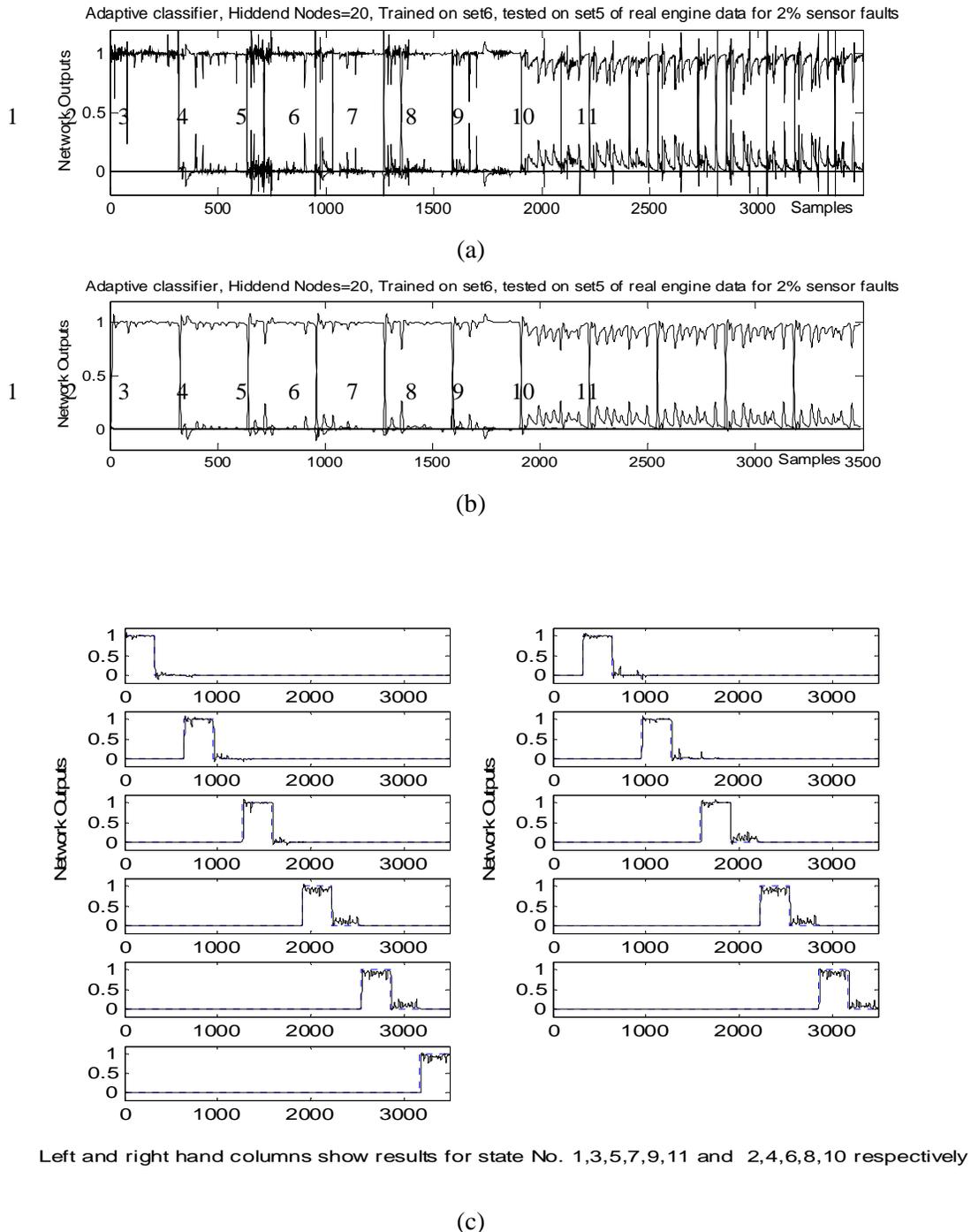
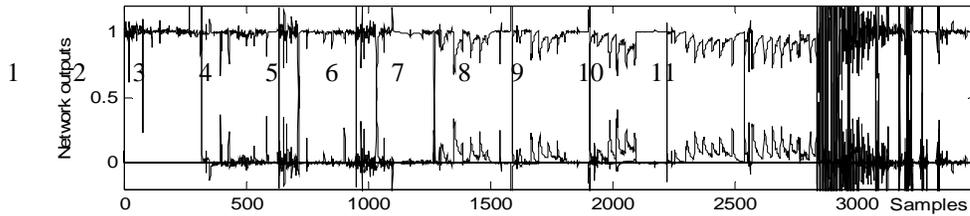


Fig. 7 Classification result for 2% faults when the network is trained on data set 6 and tested on data set 5. (a) Without data filtration (b) after low pass data filtration (c) Each state separately shown for clarity

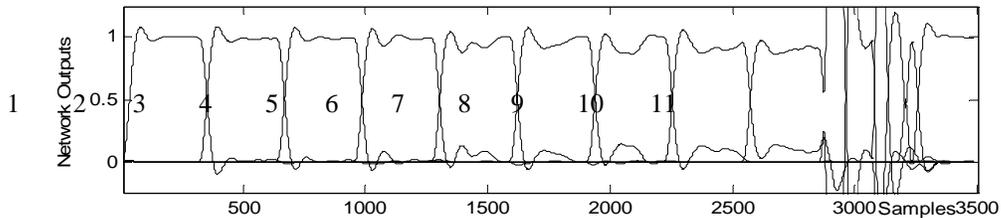
The above experimental results for the two sets of real data confirm that the dynamics issue pointed out in Section 4 is solved by adaptation of the neural classifier, though the classifier uses only current sensor outputs. To further evaluate the sensitivity and robustness of the developed adaptive classifier, a real data set with 1% faults is tested. The network is initially trained on data set 2 with 2% sensor faults and tested on data set 1 with 1% fault. The initial false alarms are high as shown in Fig. 8(a). The low pass filtering reduces the number of false alarms but some are still present (Fig. 8(b)). It is seen that a couple of misclassifications are present as indicated in Fig. 8(c). Fault diagnosis for 1% sensor faults was also tried for other sets of the real engine data and similar results were achieved. With a closer look at the test results, it was found that fault states 9 and 10 cause misclassifications.

Adaptive classifier, Hiddend Nodes=20, Trained on set2, tested on set1 of real engine data for 1% sensor faults

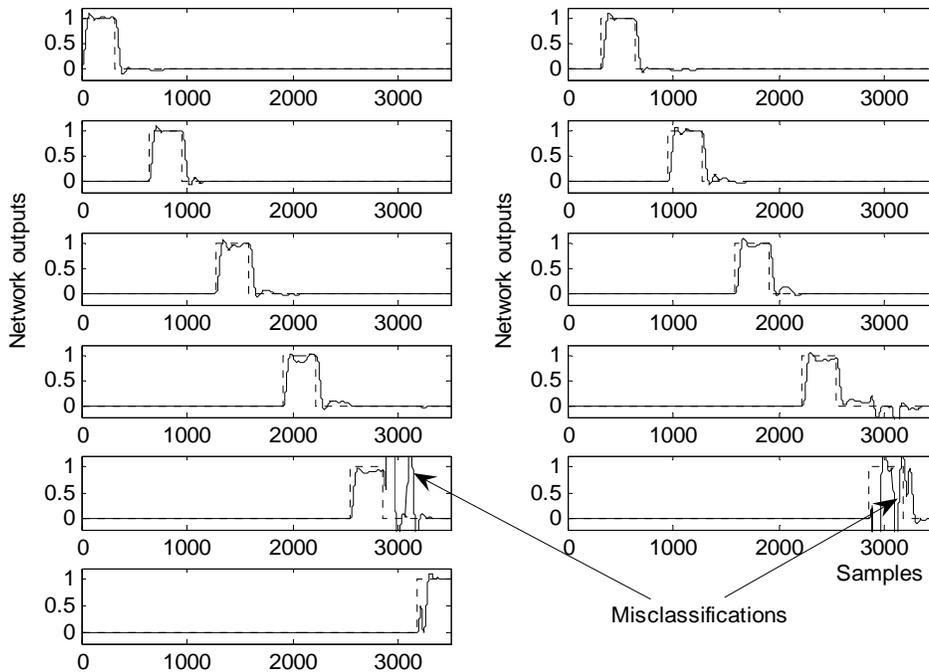


(a)

Adaptive classifier, Hiddend Nodes=20, Trained on set2, Tested on set1 of real engine data for 1% sensor faults



(b)



Left and right hand columns show results for state No. 1,3,5,7,9,11 and 2,4,6,8,10 respectively

(c)

Fig. 8 Classification result for 1% fault when the network is trained on data set 2 and tested on data set 1 (a) Without data filtration (b) after low pass data filtration (c) Each state separately shown for clarity To understand the reason for the above misclassifications, the following engine crankshaft speed dynamics could help.

$$\dot{n} = -\frac{1}{I_n} \left(P_f(p_i, n) + (P_p(p_i, n) + P_b(n)) \right) + \frac{1}{I_n} H_u \eta_i(p_i, n, \lambda) \dot{m}_f(t - \Delta\tau_d)$$

where \dot{n} is the first derivative of crankshaft speed, I is the scaled moment of inertia of the engine crankshaft and its load, $\Delta\tau_d$ is the mean injection time delay, t is time (sec), n engine speed (krpm), \dot{m}_f engine port fuel mass flow rate (kg/sec), P_i absolute manifold pressure (bar), P_f friction power (kW), P_b load power (kW), P_p pumping power (kW) and H_u fuel lower heating value (kJ/kg). From the equation it can be seen that the crankshaft speed and the moment of inertia of the engine and its load have opposite effects on engine shaft acceleration. That means when the load on the engine is high then the speed of the engine would be low if the speed does not intend to change. The faults causing misclassification are speed sensor under reading (No.9) and the torque sensor over reading (No.10). Therefore, these two faults have the same effects on the engine acceleration and are consequently difficult to be isolated compared with the other faults.

6. Conclusions

Real engine data of five sensors is acquired from a one-litre Volkswagen car engine test bed under different operating states, on which ten different faults are superimposed. A fault detection and isolation scheme using an adaptive RBF classifier is employed to diagnose faults from these real data. Experiment results confirm that the sensor faults as small as 2% are clearly detected and isolated for different data sets. The neural network classifier is on-line adapted for its widths and weights to cope with model uncertainty, time varying dynamics and environment change, so that the sensitivity of the scheme to the faults and the robustness to the uncertainty and disturbances are maintained. The developed FDI scheme has a high potential to be used in car on-board monitoring system in the near future.

Acknowledgement

The first author would like to thank Universities UK for Overseas Research Student Award Scheme (ORSAS), Faculty of Technology and Environment and School of Engineering of Liverpool John Moores University for financial support.

References

- Antory D., 2005. Fault diagnosis application in an automotive diesel engine using auto-associative neural networks, *Proc. of Int. Conf. Computational Intelligence, Modelling and Control Automation*, Vienna, Austria, Nov 28-30, pp. 109-116.
- Antory D., 2007. Application of a data-driven monitoring technique to diagnose air leaks in an automotive diesel engine: A case study, *Mechanical Systems and Signal Processing*, Vol. 21, pp. 795-808.
- Capriglione D., Liguori C., Pianese C. and Pietrosanto A., 2002. On line sensor fault detection, isolation and accommodation in automotive engines, *Instrumentation and Measurement Technology Conference*, Anchorage, AK, USA, 21-23 May, pp. 1711-1716.
- Capriglione D., Liguori C., Pianese C. and Pietrosanto, A., 2003. On line sensor fault detection, isolation and accommodation in automotive engines, *IEEE Trans. on Instrumentation and Measurement*, Vol. 52, No. 4, pp. 1182-1189.
- Capriglione D., Liguori C., Pianese C. and Pietrosanto A., Analytical redundancy for sensor fault isolation and accommodation in public transportation vehicles, *IEEE Trans. on Instrumentation and Measurement*, Vol. 53, No. 4, pp. 993-999.
- Capriglione D., Liguori C. and Pietrosanto A., 2007. Real - Time Implementation of IFDIA Scheme in Automotive Systems, *IEEE Trans. on Instrumentation and Measurement*, Vol. 56, No. 3, pp. 824-830.
- Crossman J. A., Guo H., Murphey Y. L. and Cardillo J., 2003. Automotive signal fault diagnostics – Part I: Signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection, *IEEE Trans. on Vehicular Technology*, Vol. 52, No. 4, pp. 1063-75.
- Ding E. L., Fennel H. and Ding S. X., 2004. Model-based diagnosis of sensor faults for ESP systems, *Control Engineering Practice*, 12, pp. 847-56.
- Evans-Pughe C., 2006. Learning to drive [tightening emissions regulations], *Engineering & Technology, Institution of Engineering and Technology*, UK, Vol. 1, No. 2, pp. 42-45.
- Gertler J. J., Costine M., Xiaowen F., Hira R., Kowalczuk Z., and Qiang L., 1993. Model-based on-board fault detection and diagnosis for automotive engines, *Control Engineering Practice*, Vol. 1, No. 1, pp. 3-17.
- Jakubek S. and Strasser T., 2002. Fault diagnosis using neural networks with ellipsoidal basis functions, *Proc. of American Control Conference*, May 8-10, Anchorage, AK, pp. 3846-3851.
- Kimmich F., Schwarte A. and Isermann R., 2005. Fault detection for modern diesel engines using signal and process model-based methods, *Control Engineering Practice*, Vol. 13, pp. 189-203.
- Ljung L., 1999. System Identification - Theory for the User, 2nd edition, Prentice-Hall, pp. 361-369.

- Manzie C., Palaniswami M. and Watson, H., 2001. Gaussian networks for fuel injection control, *Proc. IMechE Part D, Automobile Engineering*, pp. 1053-1068.
- Muldoon S. E., Kowalczyk M., and Shen J., 2002. Vehicle fault diagnostics using sensor fusion approach, *Proc. of IEEE*, Vol. 2, 12-14 June, pp. 1591-96.
- Nyberg, M. and Stutte, T., 2004. Model based diagnosis of the air path of an automotive diesel engine, *Control Engineering Practice*, Vol. 12, pp. 513-525.
- Sangha M. S., Yu D. L. and Gomm J. B., 2006. On-board monitoring and diagnosis for spark ignition engine air path via adaptive neural networks, *Proc. of IMechE Part D, Journal of Automobile Engineering*, Vol. 220, pp. 1641-1655.
- Tan Y. and Saif M., 2000. Neural-networks-based nonlinear dynamic modelling for automotive engines, *Neurocomputing*, 30, pp. 129-142.
- Vinsonneau J.A.F., Shields D.N., King P.J. and Burnham K.J., 2001. Fault detection and modelling for an automotive engine, *Proceedings of the 14th International Conference on Systems Science*, Vol. 3, Wroclaw, Poland, 11-14 Sept., pp. 129-136.
- Wu Jian-Da and Chen Jien-Chen, 2006. Continuous wavelet transform technique for fault signal diagnosis of internal combustion engines, *NDT & E International*, Vol. 39, No. 4, pp. 304-311, June.
- Yu D. L. and Gomm J. B., 1999. Sensor fault diagnosis in a chemical process via RBF neural network, *Control Engineering Practice*, Vol. 7, No. 1, pp. 49-55.
- Yu D. L., Chang T. K. and Yu D. W., 2005. Fault tolerant control of multivariable processes using auto-tuning PID controller, *IEEE Trans on Systems, Man and Cybernetics Part B*, Vol. 35, No. 1, pp. 32-43.
- Zhang X., Marios M., Polycarpou, and Thomas P., 2002. A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems, *IEEE Trans. on Automatic Control*, Vol. 47, No. 4, pp. 576-590.
- Vemuri A. T., 1999. Diagnosis of Sensor Bias Faults, *Proceedings of American Control Conference*, San Diego, California, June, pp. 460-464.

Biographical notes

Dr. M. S. Sangha is of Cummins Engine Co. Ltd, Royal Oak Way South, Daventry, NN11 8NU, UK

Dr. Dingli Yu received B.Eng from Harbin Civil Engineering College, China in 1982, M.Sc from Jilin University of Technology (JUT), China in 1986, and the PhD from Coventry University, U.K. in 1995, all in Control Engineering. Dr. Yu was a lecturer at JUT from 1986 to 1990, a visiting researcher at University of Salford in 1991, a post-doctoral research fellow at Liverpool John Moores University from 1995 to 1998. He joined LJMU in 1998 as a senior lecturer and was promoted to a reader in 2003, and to the professor of Control Systems in 2006. He is the associate editor for two journals, IJISS and IJMIC and, the IFAC SAVEPROCESS Committee member, and IPC member for many international conferences. His current research interests include fault detection and fault tolerant control of bilinear and nonlinear systems, adaptive neural networks and their control applications, model predictive control for chemical processes and engine systems.

Dr. J. Barry Gomm received the B. Eng. first class degree in electrical and electronic engineering in 1987 and the Ph. D. degree in process fault detection in 1991 from Liverpool John Moores University (LJMU), UK. He joined the academic staff at LJMU in 1991 and is a reader in intelligent control systems. His research interests include neural networks for modeling, control and fault diagnosis of non-linear processes, intelligent methods for control, system identification, adaptive systems, chemical process, and automotive applications.

Received December 2010

Accepted April 2011

Final acceptance in revised form July 2011