International Journal of Automotive Technology, Vol. 8, No. 5, pp. 637–650 (2007)

1229–9138/2007/036–13

ADAPTIVE FDI FOR AUTOMOTIVE ENGINE AIR PATH AND ROBUSTNESS ASSESSMENT UNDER CLOSED-LOOP CONTROL

M. S. SANGHA, D. L. YU* and J. B. GOMM

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

(Received 25 January 2007; Revised 1 August 2007)

ABSTRACT–A new on-line fault detection and isolation (FDI) scheme has been proposed for engines using an adaptive neural network classifier; this paper investigates the robustness of this scheme by evaluating in a wide range of operational modes. The neural classifier is made adaptive to cope with the significant parameter uncertainty, disturbances, and environmental changes. The developed scheme is capable of diagnosing faults in the on-line mode and can be directly implemented in an on-board diagnosis system (hardware). The robustness of the FDI for the closed-loop system with crankshaft speed feedback is investigated by testing it for a wide range of operational modes, including robustness against fixed and sinusoidal throttle angle inputs, change in load, change in an engine parameter, and all changes occurring simultaneously. The evaluations are performed using a mean value engine model (MVEM), which is a widely used benchmark model for engine control system and FDI system design. The simulation results confirm the robustness of the proposed method for various uncertainties and disturbances.

KEY WORDS : On-board fault diagnosis, Automotive engines, Adaptive neural networks, Adaptive fault classification

NOMENCLATURE

t	: time (sec)
α	: throttle plate angle (degrees)
n	: engine speed (rpm/1000)
$\dot{m}_f$	: engine port fuel mass flow (Kg/sec)
T_a	: ambient temperature (Kelvin)
p_i	: absolute manifold pressure (bar)
T_i	: intake manifold temperature (Kelvin)
$\dot{m}_{at}$	: air mass flow past throttle plate (Kg/sec)
T_{EGR}	: EGR temperature (Kelvin)
$\dot{m}_{ap}$	: air mass flow into intake port (Kg/sec)
$\dot{m}_{EGR}$	: EGR mass flow (Kg/sec)
V_i	: manifold + port passage volume (m^3)
R	: gas constant=287×10^{-5} kJ/kg/K
κ	: ratio of specific heats = 1.4 for air
I	: crankshaft load inertia (kg m^2)
P_f	: friction power (kW)
P_b	: load power (kW)
λ	: Relative value to indicate the air/fuel ratio (λ=1 corresponds to air/fuel ratio of 14.7)
P_p	: pumping power (kW)
H_l	: fuel lower heating valve (kJ/kg)
$\Delta\tau_d$	: injection torque delay time (sec)

*Corresponding author. e-mail: d.yu@ljmu.ac.uk

1. INTRODUCTION

An automotive engine is a complex machine that is controlled and monitored by a sophisticated electronic system called the electronic control unit (ECU). The need of an advanced ECU arose due to legislative requirements for pollution control. All petrol cars sold within Europe since January 1, 2001, and diesel cars manufactured from 2003, must have on-board diagnostic systems to monitor engine emissions. These systems were introduced in line with European Directive 98/69/EC (Official Journal of the European Communities, 1998) to monitor and reduce emissions from cars. All such cars must also have a standard European On-Board Diagnostics (EOBD) socket that provides access to this system. EOBD systems monitor and store information from sensors throughout the car, e.g., air flow sensors and oxygen sensors. Sensor values outside an acceptable range trigger a Diagnostic Trouble Code (DTC). New diagnostic tools help in reading and interpreting these codes; and view the live sensor output. EOBD is the European equivalent of the American On-Board Diagnostics - II (OBD-II) standard which applies to cars sold in the United States from 1996.

Some engine faults can lead to an increase in emissions and adversely affect fuel efficiency. Some serious faults can even lead to the ceasing of an engine or

even an accident. That is why fault detection, isolation, and accommodation have become so important for the automotive industry. 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, which 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 by many authors (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 by Gertler (Gertler *et al.*, 1993). The application of a data-driven monitoring technique 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 hardware-in-the-loop simulation (HILS) system was developed and performance of a commercial Electronic Stability Program (ESP) Electronic Control Unit (ECU) was evaluated for a virtual vehicle under various driving conditions (Lee *et al.*, 2007). This HILS system can be used in various applications such as benchmarking, comparison of commercial ECUs, and detection of fault and malfunction of ESP ECUs. A Kohonen network-based fault diagnosis system for fault diagnosis and monitoring of starter motors was proposed (Bay and Bayir, 2005) for the fault diagnosis of six different faults in starter motors which made it possible to diagnose the faults before they occurred by keeping fault records of past occurrences. The effectiveness of a non-linear PCA-based monitoring scheme was illustrated for the drifting fault in the fuel flow sensor due to a partial blockage of the intercooler in a Volkswagen TDI 1.9 litre diesel engine (Antory *et al.*, 2005). The pattern recognition and classification abilities of networks were applied to crankshaft speed fluctuation data for engine-fault diagnosis, and multidimensional mapping capabilities were investigated as an alternative to large lookup-tables and calibration functions (Shayler *et al.*, 2000). A continuous wavelet transform technique for the fault signal diagnosis in an IC engine and its cooling system was presented in (Wu and Chen, 2006). A neural network model-based fault classification system for a non-linear dynamic process was investigated (Yu and Gomm, 2003) and the real data experiment showed that sensor faults could be detected and isolated even without a mathematical model process. An FDI scheme for abrupt and incipient faults presented (Zhang *et al.*, 2002) using online estimators is a good example of an automated fault-diagnosis methodology.

In this paper, two components and two sensor faults with four different levels of intensities have been investigated as four typical and practical examples of SI engine faults. The faults considered are realistic and have been considered by previous authors (Nyberg and Stutte, 2004; Capriglione *et al.*, 2003). The two component faults are exhaust gas recycle (EGR) valve stoppage and gas leakage in the intake manifold. The two sensor faults are intake manifold pressure and temperature sensor faults.

In this paper, a new on-line FDI scheme proposed for engines using an adaptive neural network classifier (Sangha *et al.*, 2006) is thoroughly tested for a wide range of operational modes to check robustness of the proposed scheme. The classifier system is adaptive to cope with the significant parameter uncertainty, disturbance, and environment change. It is capable of on-line fault diagnosis which can be directly implemented in an on-board diagnosis system (hardware). During operation, the network classifier learns parameter changes in the engine due to aging or environmental change. It can also adapt to engine-to-engine differences within a batch of products. Gaussian radial basis function (RBF) neural nets are used for this purpose and both weights and widths are adapted on-line. Every sample of engine data is first tested for a fault and then used to update the neural network. The proposed approach is applied to diagnose some simulated faults in an SI engine air path. It is impracticable for the authors to get real faulty data from a running engine at a specific time and situation. Therefore, an engine simulation model is used for fault simulation. The adaptive algorithm is also compared with a non-adaptive algorithm. Furthermore, the robustness of the developed adaptive system is investigated by testing it for a wide range of operational modes for a real automotive engine running on road viz. change in speed set-point, load, and the engine parameter. The nobility of this paper consists in the successful demonstration of robustness of the developed adaptive neural network-based FDI algorithm.

The developed fault diagnosis scheme directly contributes in improving the engine performance. A fast and reliable fault diagnosis would result in a quick realization of the problem. For example, a gas leakage in the manifold would lead to increased air pollution and degraded engine performance. Detection of faults in time can also prevent possible catastrophe from developing further. Similarly, the EGR valve fault will adversely affect the engine performance. If the EGR valve is clogged, the engine performance will go down and an early detection of the fault is required for engine maintenance and repair. Some sensor faults that are used for feedback control will affect both dynamic and steady-state performance.

2. FAULT DIAGNOSIS METHOD

According to the engine air path dynamics, four variables are chosen as the network inputs: the throttle angle, the

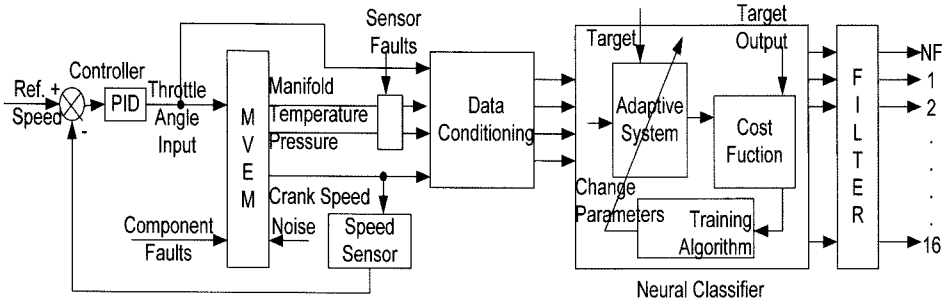

Figure 1. Information flow of the fault diagnosis.

manifold pressure, the manifold temperature, and the crankshaft speed. The RBF network, as the fault classifier, will receive all possible and relevant signals containing fault information, and has 17 outputs each indicating one of the investigated states: one for no-fault and 16 for 16 faults. The information flow for the fault diagnosis is illustrated in Figure 1.

The feedback system in a real automotive is a human element (the driver). The speed is attempted to be kept constant by adjusting the throttle angle. In the model, the human controller is represented by a PID controller. The mean value engine model (MVEM) receives a controlled throttle angle input. Component faults are simulated in the model one by one and an appropriate level of measurement noise is added to all input and output measurements. All four inputs and outputs are conditioned, normalised, and fed to the adaptive classifier. Widths in the hidden nodes and the weights in the output layer of the RBF network are adapted to minimise the sum squared error between the output from the adaptive system and the pre-decided target output. The gradient Descent method is used for the widths of the RBF network. The width in each hidden layer node is usually 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 be on-line to 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:

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

where $x(k)$ is the network input vector at iteration k, c_i is the centre of the i^{th} activation function, $\phi_i(k)$ is the Gaussian basis activation function, $w_{ij}(k)$ is the output layer weight element connecting the j^{th} hidden node to the i^{th} output, α is a learning factor with $0<\alpha<1$. The complete mathematical derivation of the above equation can be found in (Sangha *et al.*, 2006)

While the fault classifier diagnoses faults on-board, the classifier is adapted on-line so that the model-plant mismatch, parameter uncertainty and especially the time varying dynamics caused by mechanical wear of components and environment change can be modelled. In this way the classification error and consequently the false alarms will be greatly reduced. Here the false alarm is the alarm 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 are not considered by the fixed parameter classifier.

The fault classification and on-line adaptation are implemented as follows. First, 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 as previously described. This is followed by adaptation of the weights using the recursive least squares (RLS) algorithm (Ljung, 1999).

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

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

and a fault is believed to be fired when

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

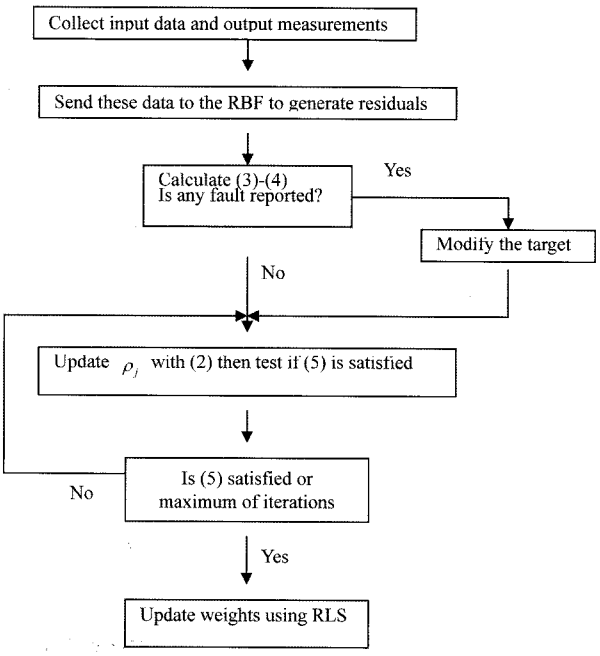

Figure 2. Flow chart of fault diagnosis and classifier updating.

where k is the sample instant, r_j is the residual and r_t is the 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. It was found that a single iteration updating with the gradient descent method would not reach the minimum if the learning rate is chosen small, while a large learning rate will cause unstable convergence.

The recursive updating of the widths runs until the following is satisfied,

$$\frac{\partial J}{\partial \rho_i} \leq \sigma, \quad i=1, \cdots, n_h \tag{5}$$

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 Figure 2.

3. ENGINE DYNAMICS AND CONTROLLER DESIGN

3.1. Mean Value Engine Model

A mean value engine model (MVEM) is chosen for fault simulation as well as testing. A speed feedback loop along with a PID controller is added to the MVEM in this research as shown in Figure 3. In a real automotive the speed feedback control and accordingly the manipulated variable, throttle angle, is given by a human element (driver). Here however, the fault detection and evaluation are done when the engine is under closed-loop speed control to simulate the real world situation. The modified MVEM has reference speed as the input and four outputs: throttle position, intake manifold temperature, intake manifold pressure, and crank shaft speed, respectively.

To investigate the feasibility of the developed method under closed-loop speed control, the dynamics of the modified and controlled MVEM is introduced. It consists of three sub-models that describe the intake manifold dynamics including air mass flow, pressure and temperature and the crankshaft speed. The engine dynamics are explained briefly in the following sections.

3.2. Manifold Filling Dynamics

The engine air path is schematically illustrated in Figure 4. Its dynamics are briefly presented as follows and the

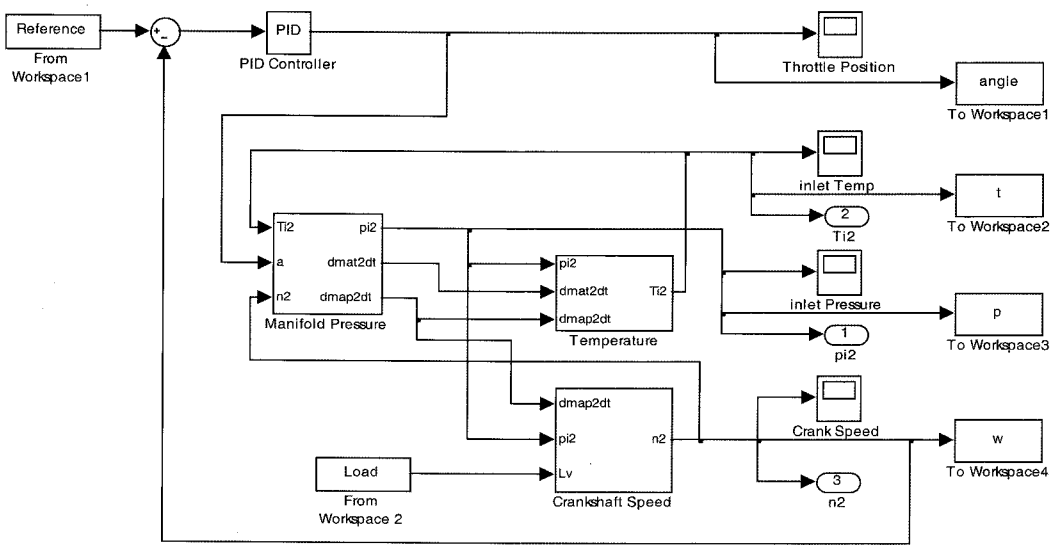

Figure 3. Simulink model of MVEM with crankshaft speed feedback.

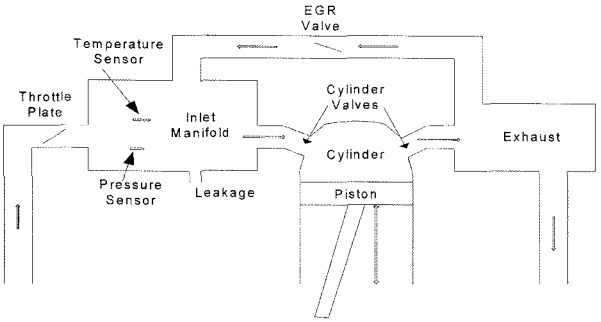

Figure 4. Schematic of the air intake and exhaust system of the engine.

physical parameters are defined in the nomenclature. The details of the dynamics can be referred to in (Handricks *et al.*, 2000).

The manifold filling dynamics in reality is based on an adiabatic operation rather than an isothermal one. The manifold pressure can be represented as

$$\dot{p}_i=\frac{\kappa R}{V_i}(-\dot{m}_{ap}T_i+\dot{m}_{at}T_a+\dot{m}_{EGR}T_{EGR}) \quad (6)$$

where the dot on the top of a variable, such as $\dot{p}_i$, denotes the first derivative of the variable with respect to time. The positive terms within the brackets show the in-flow of gas and the negative term shows the outflow of gas from the intake manifold (Figure 4). Using the law of energy conservation, a state equation which describes the time development of the intake manifold temperature can be given as,

$$\dot{T}_i=\frac{RT_i}{p_iV_i}[-\dot{m}_{ap}(\kappa-1)T_i+\dot{m}_{at}(\kappa T_a-T_i)+\dot{m}_{EGR}(\kappa T_{EGR}-T_i)] \quad (7)$$

3.3. Crank Shaft Speed Dynamics

Applying the law of conservation of rotational energy, the crankshaft dynamics of an SI engine MVEM are described by equation (8).

$$n=-\frac{1}{In}P_f(p_i,n)+P_p(p_i,n)+P_b(n)+\frac{1}{In}H_l\eta_i(p_i,n,\lambda)\dot{m}_f(t-\Delta\tau_d) \quad (8)$$

Where I is the scaled moment of inertia of the engine and its load and the mean injection/torque time delay have been taken into account with variable $\Delta\tau_d$. λ=1 corresponds to the air/fuel ratio (AFR) of 14.7 for gasoline and 14.5 for diesel. At λ=1 we have stoichiometry or the point at which the most complete combustion takes place. λ gives a measure of AFR, which is independent of the type of fuel being used. $\lambda > 1.0$ indicates excess air (lean mixture), while $\lambda < 1.0$ indicates excess fuel (rich mixture).

3.4. Controller Design

A simple closed-loop PID controller is shown in Figure 5. The variable 'e' represents the tracking error, the difference between the desired input value (reference signal) 'R', and the actual output 'Y'. This error signal 'e' is sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The controller output signal 'u' is equal to the proportional gain K_p times the magnitude of the error plus the integral gain K_i times the integral of the error plus the derivative gain K_d times the derivative of the error,

$$u(t)=K_pe(t)+K_i\int e(t)dt+K_d\frac{de(t)}{dt}$$

This signal 'u' is put into the MVEM, completing the feedback loop fed back to the reference. The well known Ziegler-Nichols method is used for tuning the PID controller. Initially, K_i and K_d gains are set to zero. The proportional gain is increased until it reaches the critical-gain K_c at which the output of the loop starts to oscillate. K_c and the oscillation period P_c are used to set the gains as K_p=0.45*K_c and K_i=1.2*K_p/P_c. The desired output is achieved without the use of the derivative gain. Therefore, the derivative gain is kept zero to keep the controller as simple as possible.

A set of five random values in the range of 2 to 4 kRPM are applied as reference signals. Each random speed is sustained for 6 seconds before the speed signal is changed to the next value because the outputs of the simulation reach their steady state values in six seconds. The data is sampled every 0.5 seconds. Therefore, 12 data points are collected in every six seconds of time. The output response of the crankshaft speed for the no fault case for five different reference signals is shown in Figure 6. The output crankshaft speed follows the input reference speed without much overshoot, delay time, and zero steady-state error. The chosen PID (K_p=10, K_i=10, K_d=0) settings give an acceptable level of performance of the controller for further experimentation.

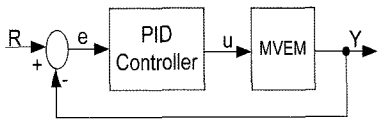

Figure 5. Closed-loop PID controller.

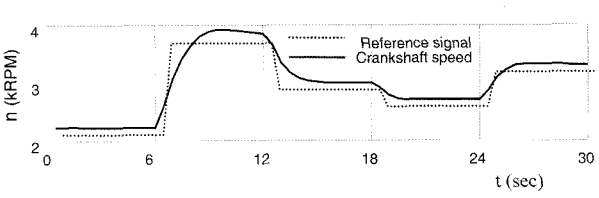

Figure 6. No fault outputs for five random speed reference signals.

4. FAULT DIAGNOSIS

4.1. Simulating Faults

The sensor faults can occur due to two reasons:
(i) Aging and wear & tear of the mechanical parts of the deflection meter and
(ii) Electrical fault, e.g. short circuit or open circuit fault in the signal cable.
The electrical faults are easy to detect because open circuit and short circuit faults will cause a full deflection or zero deflection in the meter, respectively. On the contrary, the aging and mechanical faults cause an incorrect meter reading i.e. over- or under- reading of the actual values. Both cases of under- and over- reading of the temperature or pressure measurements are considered here. Air leakage in the air path can happen due to the following reasons:
(i) missing gas caps
(ii) loose gas caps
(iii) leaks in gas caps or vapour vent lines
(iv) disconnected purge lines (Reineman, 2000).
Current OBD regulations require monitoring of any leaks (for the 2003 year model and after) that exceed 0.02 inches in diameter (0.5 mm approx). It is not practical to create some component faults in a running engine in real life, such as air leakage in the manifold or EGR valve stoppage. Therefore the faults are simulated in a Matlab engine model in this research. The air leakage is simulated in the modified mean value engine model as a percentage of the total air mass flow in the intake manifold explained later. The EGR valve can be stuck up in any position where there is a failure of the EGR valve positioning control. This will lead to a fixed percentage EGR flow through the valve. There can be many reasons for failure of the EGR valve positioning system and which have not been investigated in this paper. The investigation is focused on the detection and isolation of the fault and its intensity and not on pin-pointing the actual component failure of the EGR system viz.
(i) EGR open circuit fault
(ii) EGR vent solenoid fault
(iii) EGR step motor 1 fault
(iv) EGR step motor 2 fault
(v) EGR vacuum regulator fault
(vi) EGR boost solenoid control fault etc.
Details of the simulation of the faults are described ahead.

4.1.1. No fault

For the no fault situation, the EGR is assumed to be 1/6 (16.67%) of the total air mass flow in the intake manifold. Practically, the EGR in a car can be as high as 20% of the total air mass flow. It is also assumed that all the sensors are working well and there is no leakage in the intake manifold.

4.1.2. Air leakage fault

To collect the engine data subjected to the air leakage fault, equation (6) is modified

$$\dot{P}_i = \frac{\kappa R}{V_i}(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR} - \Delta l) \tag{9}$$

where Δl is used to simulate the leakage from the air manifold, which is subtracted to increase the air outflow from the intake manifold. The air leakage levels are simulated as 5%, 10%, 15% and 20% of the total air intake in the intake manifold, respectively.

4.1.3. EGR valve faults

The normal value of EGR is about 16.67% of the total air mass flow, which is a realistic value of EGR feedback chosen for the experiments. The value of $\dot{m}_{EGR}$ for different fault intensities is regulated as 0%, 25%, 50%, 75% and 100% of the total EGR air mass flow. Where 0% EGR air mass flow corresponds to the complete stoppage of the EGR valve and 100% corresponds to full EGR air mass flow, i.e. the no fault condition.

4.1.4. Temperature/Pressure sensor faults

Temperature and pressure sensor faults are considered in four different intensities: Sensors over-reading 20% or 10% and sensors under-reading 10% or 20% of the normal value. The faulty data for the sensors is generated using multiplying factors (MFs) of 1.2, 1.1, 0.9 and 0.8

Table 1. All the 17 states and multiplying factors (MFs).

S.N	Fault Name	MFs
1	No Fault (NF)	
2	5% Leakage in Intake Manifold	
3	10% Leakage in Intake Manifold	
4	15% Leakage in Intake Manifold	
5	20% Leakage in Intake Manifold	
6	EGR valve stuck 25% closed	
7	EGR valve stuck 50% closed	
8	EGR valve stuck 75% closed	
9	EGR valve stuck 100% closed	
10	Temp. sensor 20% over reading	MF=1.2
11	Temp. sensor 10% over reading	MF=1.1
12	Temp. sensor 10% under reading	MF=0.9
13	Temp. sensor 20% under reading	MF=0.8
14	Pressure sensor 10% over reading	MF=1.2
15	Pressure sensor 20% over reading	MF=1.1
16	Pressure sensor 20% under reading	MF=0.9
17	Pressure sensor 10% under reading	MF=0.8

for over or under reading, respectively, as shown in Table 1.

4.2. Network Training

Two RBF networks are used for fault classification, with one for the non-adaptive classifier and the other for the adaptive classifier. Both networks have the same structure and will be trained with the same training data as well as using the same training algorithm. The training for the adaptive network is referred to as initial training. After training, the non-adaptive network will be used to do fault diagnosis with the test data without on-line training, while the adaptive network will be used with the same test data but with on-line training. This establishes a fair basis for comparison between the adaptive and non-adaptive classifiers.

The network input variables are chosen according to the experience in engine modelling as the four variables shown in Figure 1: throttle angle, manifold pressure, manifold temperature and crankshaft speed. Therefore, the network has 4 inputs. Each network output is used to indicate the occurrence of one faulty state 0 (zero), which implies that the fault does not occur while 1 (one) implies that the fault occurs. Therefore, the network has 17 outputs with each corresponding to the one fault or no-fault condition. Twenty data sets for different initial and final throttle angle positions are collected as shown in Table 2.

Two data sets, one acceleration data with θ=26°→34° and one deceleration data with θ=34°→26°, are used as test data. All the remaining 18 sets are used as training data. As each training data set has the same pattern for 17 faults, one training target matrix X_o (Figure 7) is formed and used for all the training data sets. X_o has 204 rows and 17 columns. Its first column has ones from the first row to 12th row with all other entries as zeros and the second column has ones from the 13th row to the 24th row with all other entries as zeros, and the last column has ones from the 193rd row to the 204th row with all other entries as zeros.

Table 2. Details of data sets collected for training and testing of RBF networks.

Start degree of θ	Accelerating	Decelerating	No. of data sets
22	26, 30, 34, 38	–	4
26	30, 34, 38	22	4
30	34, 38	26, 22	4
34	38	30, 26, 22	4
38	–	34, 30, 26, 22	4

Row Numbers X_o

$$\begin{matrix} 1 \sim 12 \\ 13 \sim 24 \\ \vdots \\ \\ \\ 193 \sim 204 \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & & & \\ 0 & \vdots & \ddots & & & \\ 0 & & & \ddots & & \\ 0 & & & & 1 & \\ \vdots & & & & & \ddots \end{bmatrix}$$

Figure 7. Target matrix X_o.

Thus, the transposition of the i^{th} row in X_o is used as the training target vector for the i^{th} training pattern. The centres are chosen using the K-means clustering algorithm from the training data sets. The widths were chosen using the P-nearest neighbour's algorithm. The weights were trained using the RLS algorithm. Two levels, 0 and 1, are used as the output targets of the classifier. Thus, the target matrix is a unity diagonal matrix of dimension 17 (when there is one training pattern for each fault) with each column being used as the classifier-training target vector. A successfully trained network will therefore diagnose the fault intensity as well as the fault type.

4.3. Fault Classification

Both adaptive and non-adaptive networks are used to diagnose faults with test data sets after training with the training data sets. The fault detection threshold in (4) was chosen as r_t=0.5. High thresholds may lead to missed detections while low thresholds will cause more false alarms. Mathematically, r_t should be a little bit higher than 0.5 according to the level of noise in the testing data. However, r_t=0.5 is found as a good compromise between reliability of detection and insensitivity to noise. M in (3) is chosen as 3. The averaged residual will be greatly reduced and the false alarm is consequently reduced. The threshold for the gradient of the objective function in (5) was chosen as σ=0.00001. The forgetting factor for the RLS algorithm was chosen as a constant value of λ=0.99.

The three different reference signals 2.5 kRPM, 3.0 kRPM, and 3.5 kRPM are chosen as Ref1, Ref2 and Ref3 for the speed control, respectively. No fault and faulty data is collected for all three reference signals. Both the non-adaptive and the adaptive RBF neural network classifiers are then trained and tested for six different sets of data. The results for training the networks on the Ref 1 and testing on the Ref 3 data are shown in Figure 8 and Figure 9. The number of centres for the adaptive and non-adaptive networks is chosen as 100. It is clear that the non-adaptive classifier is not able to classify the simulated faults while the adaptive network classifies the faults with just a few peak values that may cause false alarms when 0.5 is used as the fault detection threshold. These faults are classified when the engine is under closed-loop speed control.

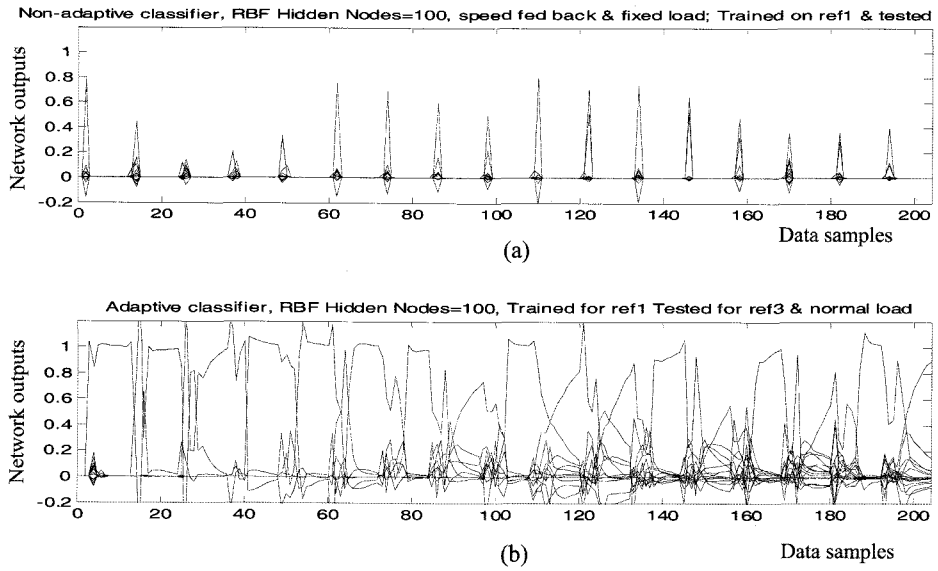

Figure 8. Networks trained on Ref1 and tested on Ref3: (a) Non-adaptive; (b) Adaptive.

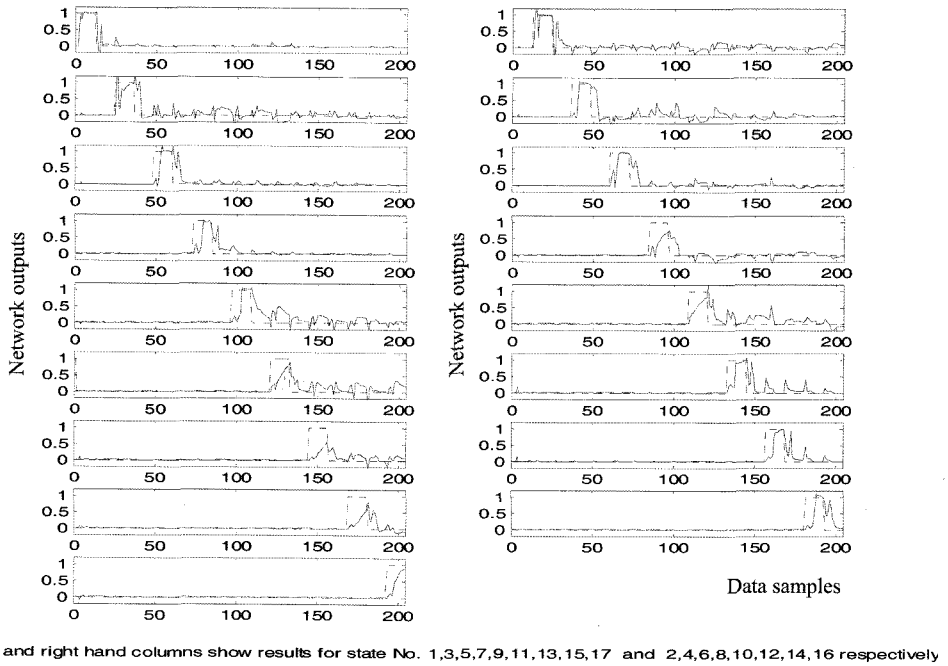

Figure 9. Details of each fault classification in Figure 8(b) shown separately for clarity.

In comparison, it is found that the adaptive classifier has performed far better than the non-adaptive classifier. Unlike the non-adaptive classifier, the adaptive classifier is able to identify all the faults but with false alarms. For clarity, Figure 8(b) is shown in an expanded form in Figure 9 with every fault classification shown separately. It can be seen that state 1 has one false alarm, state 2 has two false alarms, state 3 has one false alarm, and so on. Here the requirement of data filtration is because of the false alarms.

5. ROBUSTNESS ASSESSMENT OF FDI SYSTEM

Further to introducing speed feedback control, robustness assessment of the FDI system is carried out in the following three different modes in increasing generality of engine operation:

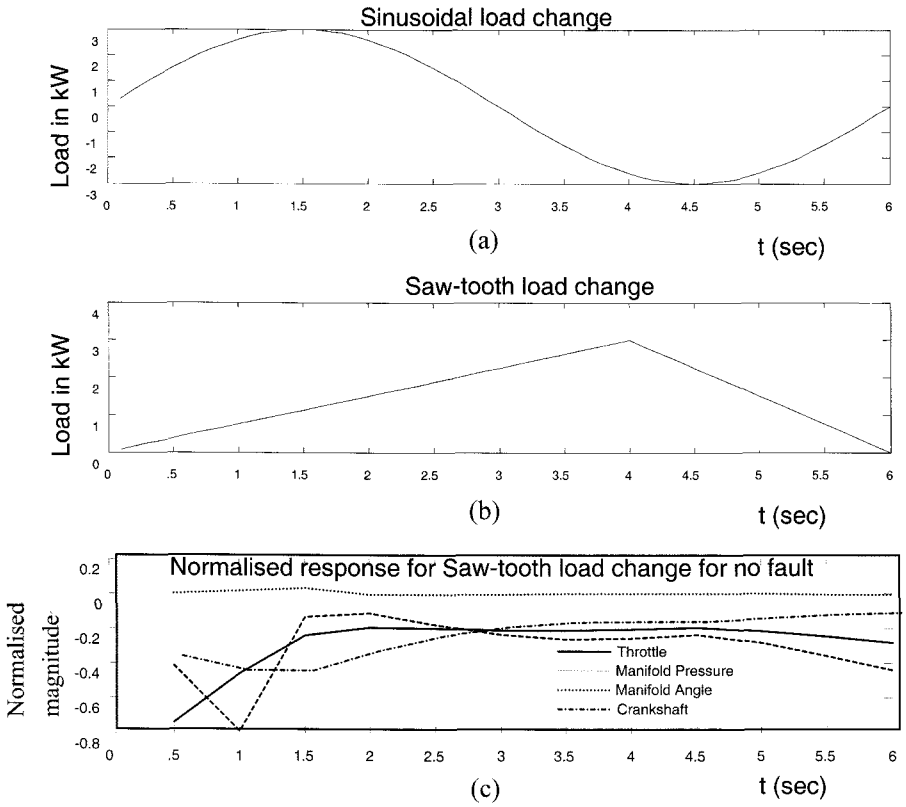

Figure 10. (a) Sinusoidal load change; (b) Saw-tooth load change; (c) No fault response for Saw-tooth load change.

(1) Load change
(2) Engine parameter change
(3) All the changes happening simultaneously

5.1. Load Change

To incorporate the provision for engine load change in the MVEM, the crankshaft speed sub-model is modified. The pumping power P_p and the friction power P_f are functions of absolute manifold pressure P_i and crankshaft speed n whereas the load power P_b is only a function of crankshaft speed as shown in equation (8). The load factor K_b (=0.47) is a constant. Engine load can be changed by changing load power P_b. The load on the engine in kW is given as:

$$\text{Engine Load}=K_b*n^3$$

Engine load is equal to the load power of engine and therefore,

$$P_b=K_b*n^3$$

Load Power in the modified model is presented as

$$P_b=K_b*n^3+L_v$$

where L_v is the load variation in kW and n is the crankshaft speed in kRPM. The reference signal is kept fixed but the load on the engine is changed in sinusoidal and saw-tooth style as shown in Figure 10(a) and (b). The change in load is applied through variable L_v as shown in the simulation diagram in Figure 2. In the case of a sinusoidal load change, the load on the automobile (engine) can be negative for some time. This represents the downhill running of the vehicle. Similarly, an increase in the load represents the uphill running of the vehicle.

Two sets of data are collected for no fault and faulty conditions; the first set of data for the sinusoidal change in the load and the second set for the saw-tooth change in the load. The reference input signal is kept constant at 2.5 kRPM for both data sets. First of all, both networks are trained with data for the sinusoidal load change and tested with data for the saw-tooth load change and then vice-versa. With both training data sets the classification results were found satisfactory. The classification test results for both classifiers when tested for the sinusoidal load change are shown in Figure 11. The results of the non-adaptive classifier are not good as shown in Figure 11(a) and it is not able to identify different faults, where as the adaptive classifier is able to identify all the faults as shown in Figure 11(b) and (c) but with false alarms. There are several false alarms in all and they can be seen in Figure 11(c) as small spikes exceeding a threshold of 0.5.

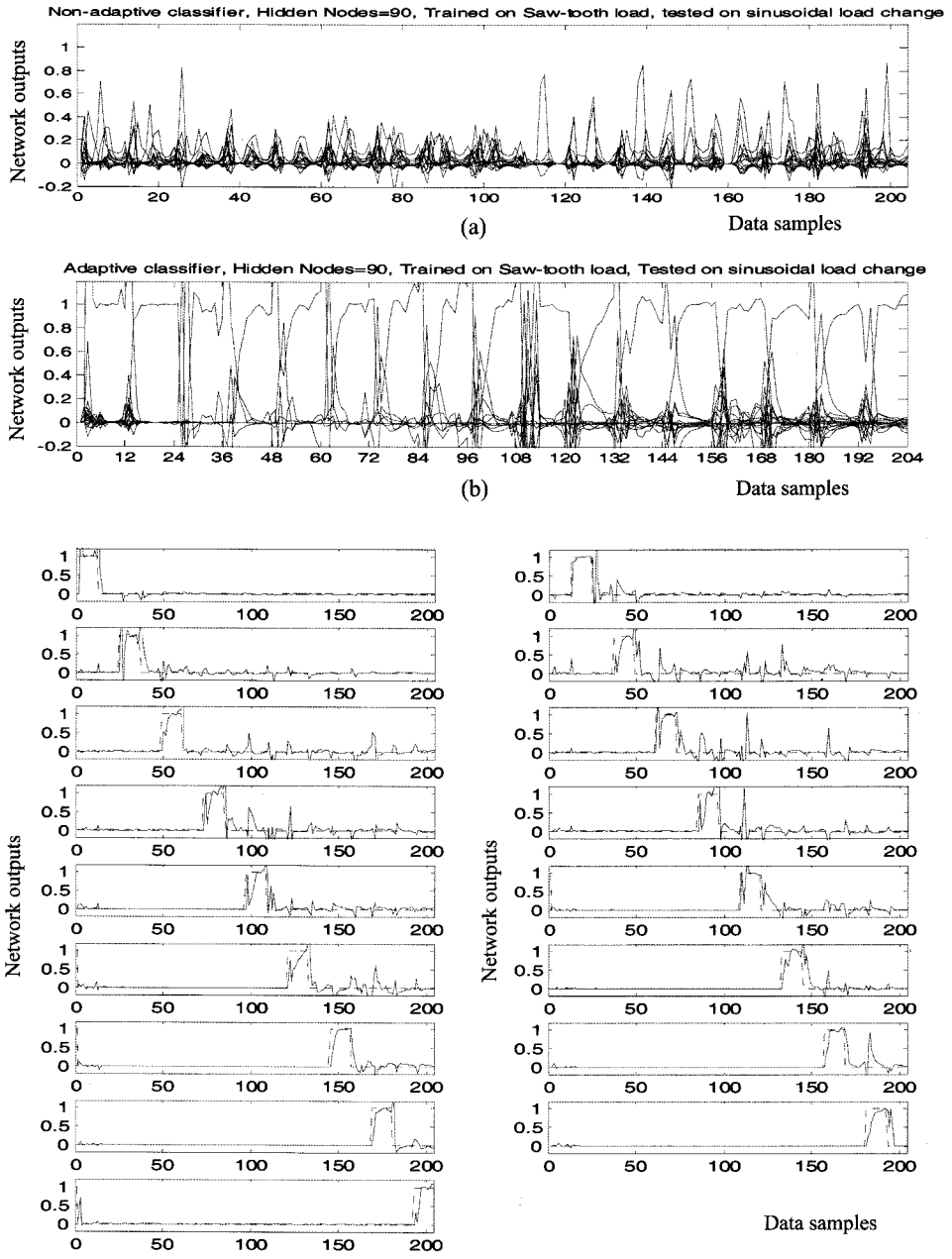

Figure 11. Networks trained on saw-tooth load and tested on sinusoidal load. (a) Results for Non-adaptive classifier; (b) Results for Adaptive classifier; (c) Details of each fault classification in; (b) shown separately for clarity.

5.2. Engine Parameter Change

The engine displacement is a constant for an engine and is 1.275 litres for the MVEM. After a few years of operation, the engine displacement has a tendency to increase by a small amount due to abrasion. In order to check the robustness of the classifier against such an aging effect of the engine, the no fault and faulty data for 1% increased engine displacement (i.e. 1.01*1.275 litres) is collected. Both classifiers are trained for the normal engine data and then tested on the data from the increased engine displacement. In this part of the experiment the speed reference signal and the load on the engine are not changed. It is found that the performance of the adaptive classifier Figure 12(b) and (c) is much better than the

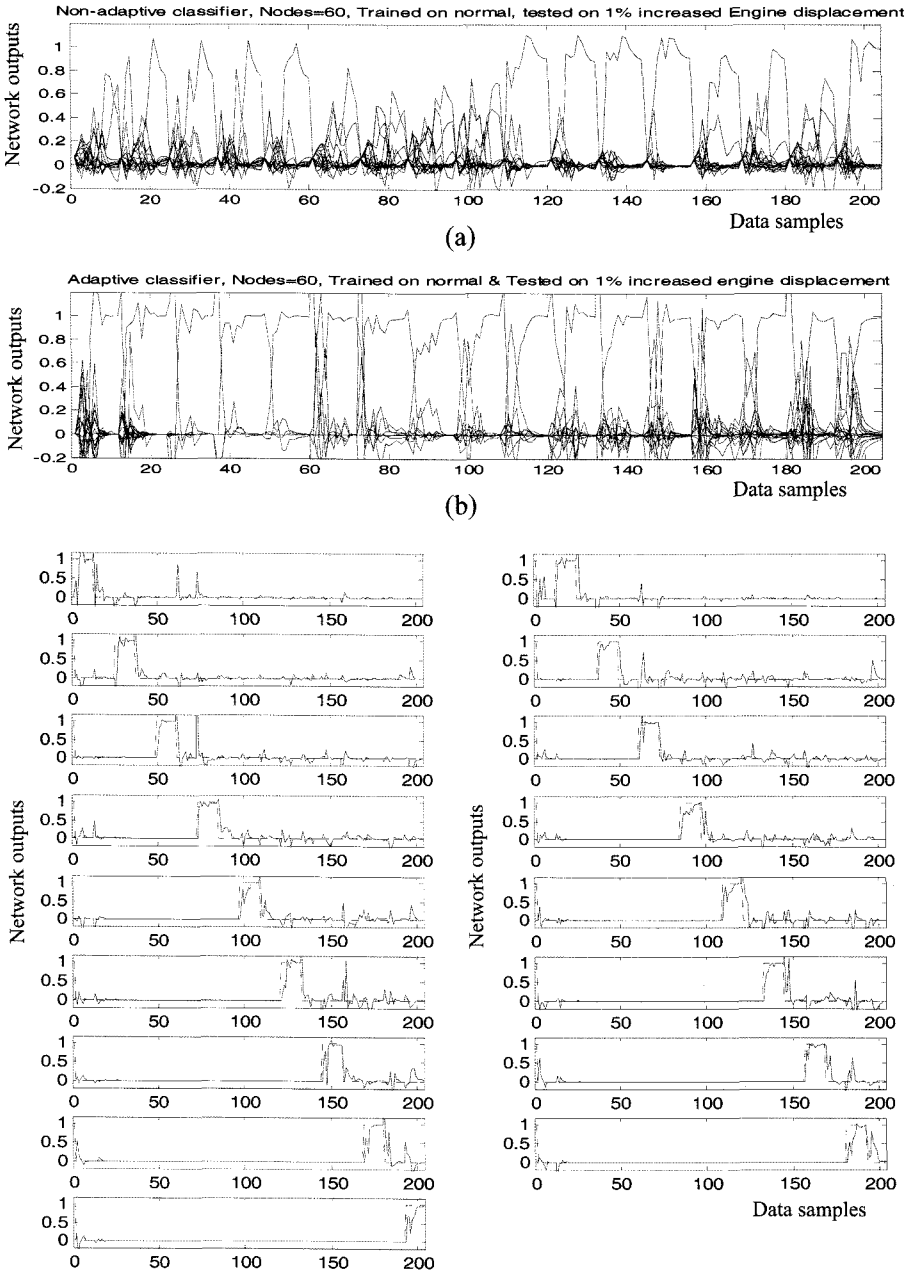

Figure 12. Networks trained on normal and tested on 1% increased engine displacement: (a) Non-adaptive classifier; (b) Adaptive classifier; (c) Details of each fault classification in b shown separately for clarity.

non-adaptive classifier Figure 12(a) but with false alarms. There are several false alarms in all and they can be seen in Figure 12(c) as small spikes exceeding the threshold of 0.5.

5.3. All the Changes Happening Together

In this section data is collected for all the changes happening simultaneously, i.e. when the reference is a saw-tooth signal, the load is changing in the sinusoidal style and the engine displacement is increased by 1%. Both non-adaptive and adaptive classifiers are trained on data with fixed reference, no change in load, no increase in engine displacement, and then tested with the data when all the changes happen together.

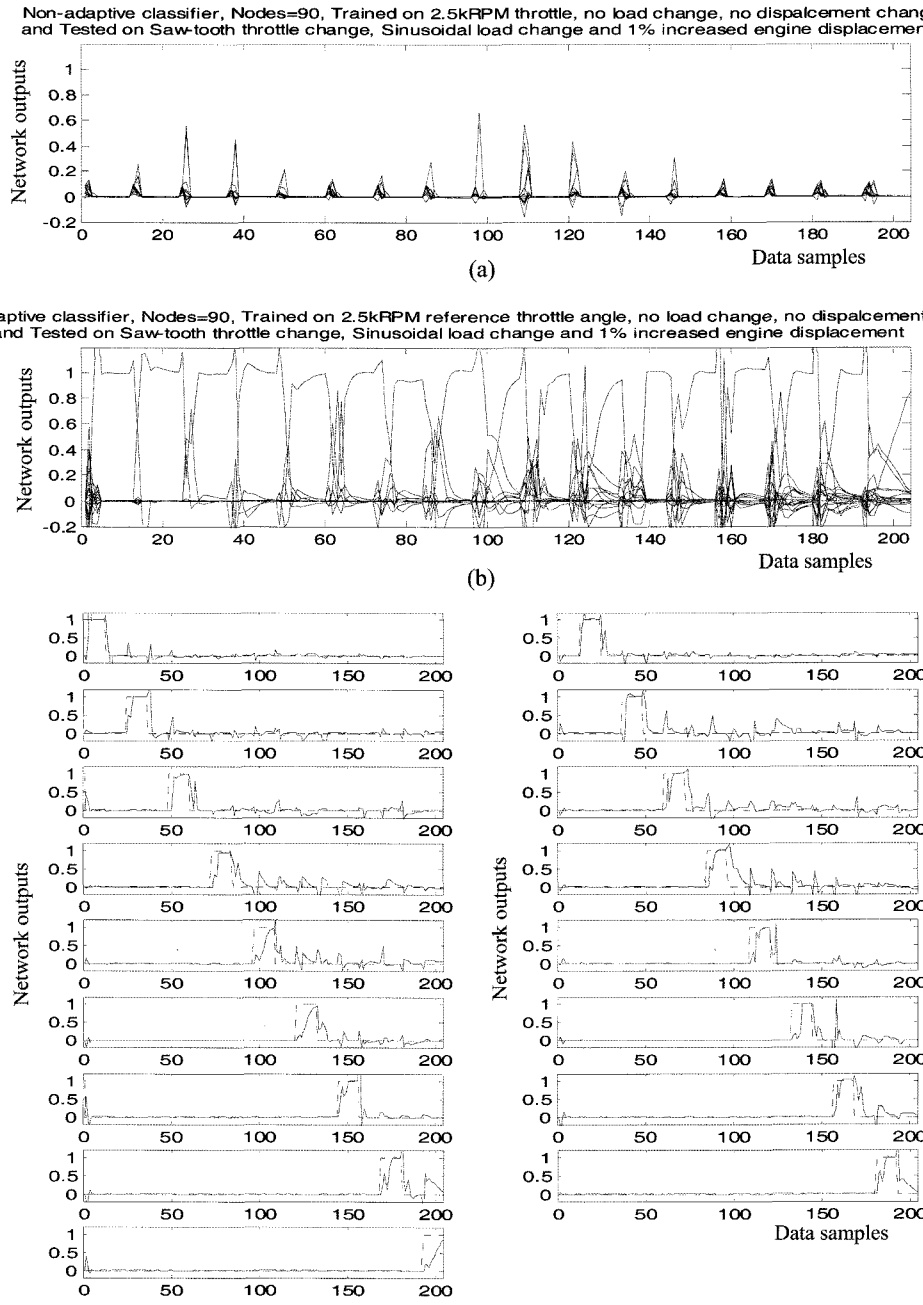

Figure 13. Networks trained on fixed reference and tested on saw-tooth reference, sinusoidal load change and 1% increased engine displacement: (a) Non-adaptive; (b) Adaptive classifier; (c) Details of each fault classification in b shown separately for clarity.

The results in Figure 13(b) and (c) show that the adaptive classifier performs well as compared to the non-adaptive classifier in Figure 13(a) but with false alarms. There are several false alarms in all and they can be seen in Figure 13(c) as small spikes going over threshold of 0.5.

In order to improve the problem of false alarms, the signal processing toolbox in Matlab is utilised and a third order low-pass digital filter is designed to suppress the spikes in the resultant data. A Butterworth digital filter

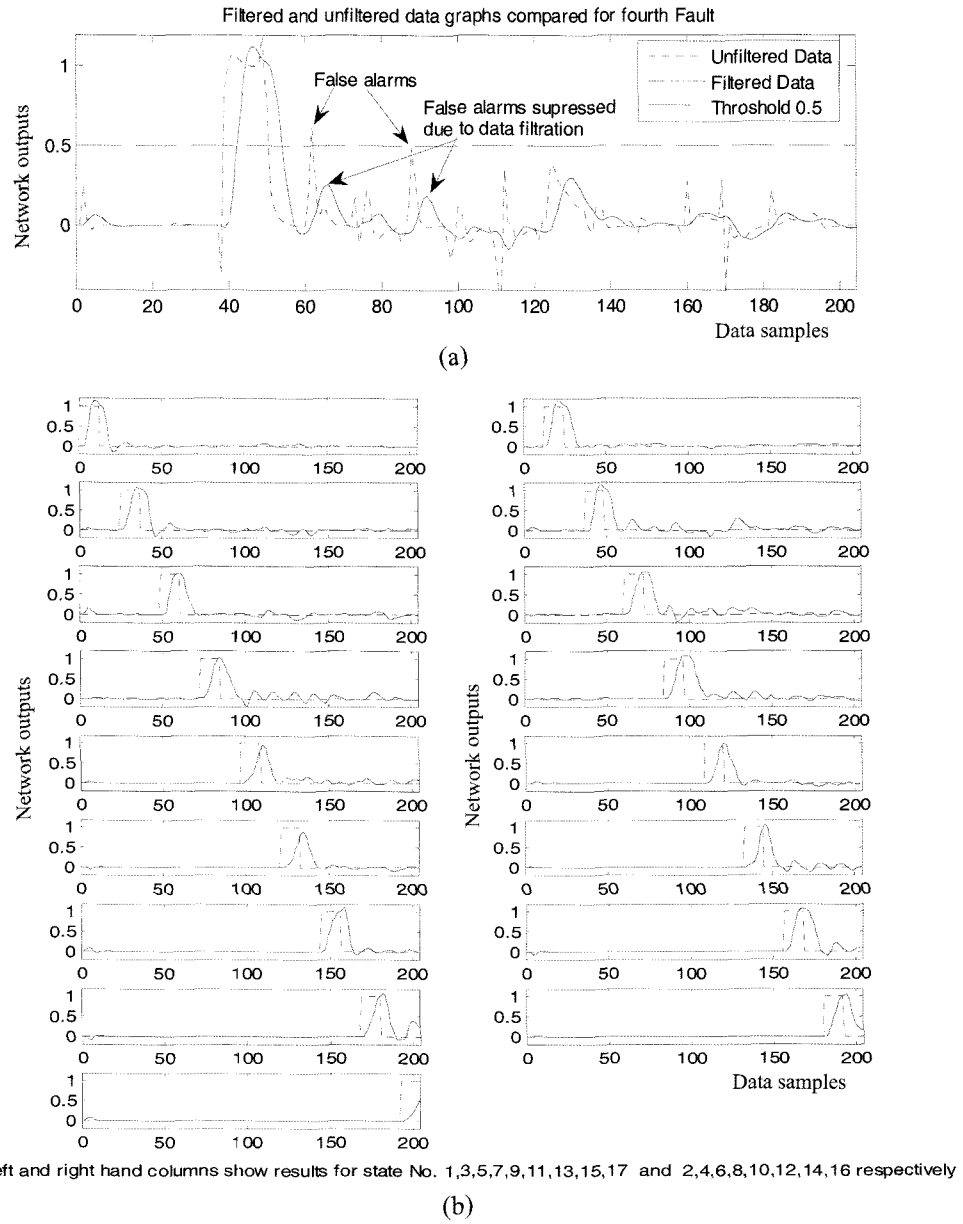

Figure 14. Classification results after using a low-pass Butterworth filter. (a) Comparison of classification results before and after filtration for fault number 4; (b) Classification results of 13; (c) after filtration.

can be designed using the Matlab function "butter". The function has two arguments N and W_n, for the order of the filter and cut-off frequency, respectively. The function returns the filter coefficients in length N+1 vectors B and A, numerator and denominator, respectively. The cut-off frequency must be $0.0 < W_n < 1.0$, with 1.0 corresponding to half the sample rate. The value of W_n is to be chosen carefully. A high value may not do any filtration at all whereas a very low value may cause a long time delay and poor filtration. A value of 0.1 has been carefully chosen for W_n which reduces the spikes to half of their original height (i.e. much below the threshold of 0.5) and causes little time delay. The false alarms are practically reduced to zero times as shown in Figure 14(a) and (b).

It can be seen from graphs in Figure 14 that the spikes causing false alarms have been filtered out and make the classification more robust and reliable.

6. CONCLUSIONS

A new adaptive RBF-based FDI method for an SI engine is evaluated for robustness. The classifier is adapted for

its widths and weights to learn changes in the system dynamics and environment. The robustness of the system is investigated for a wide range of operational modes in increasing generality. Robustness assessment has been carried out against fixed and sinusoidal throttle angle inputs, change in load, change in engine parameter, and all these changes occurring at the same time for both adaptive and non-adaptive networks. The adaptive network performs very well and the simulation test results are satisfactory for all sixteen of the faults considered. The non-adaptive classifier fails to cope up with the load change, parameter change, etc. Thus, it is not robust whereas the adaptive network classifies all the faults correctly and the false alarm is reduced to zero by the use of a low-pass filter.

Robustness assessment against different types of unknown faults and simultaneously occurring multi-faults is considered for the future work.

ACKNOWLEDGEMENT–The first author would like to thank the ORSAS of the U.K. for financial support.

REFERENCES

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,* **21**, 795–808.

Antory, D., Kruger, U., Irwin, G., and McCullough, G. (2005). Fault diagnosis in internal combustion engines using non-linear multivariate statistics. *Proc. Institution of Mechanical Engineers Part I - J. Systems and Control Engineering* **219**, **14**, 243–258.

Bay, O. F., and Bayir, R. (2005). Kohonen network-based fault diagnosis and condition monitoring of pre-engaged starter motors. *Int. J. Automotive Technology* **6**, **4**, 341–350.

Capriglione, D., Liguori, C., Pianese, C., and Pietrosanto, A. (2003). On line sensor fault detection, isolation and accommodation in automotive engines. *IEEE Trans. Instrumentation and Measurement* **52**, **4**, 1182–1189.

Evans-Pughe, C. (2006). Learning to drive [tightening emissions regulations], engineering & technology. *Institution of Engineering and Technology* **1**, **2**, 42–45. UK.

Gertler, J. J., Costine, M., Xiaowen, Fang, Hira, R., Kowalczuk, Z., and Qiang, Luo (1993). Model-based on-board fault detection and diagnosis for automotive engines. *Control Engineering Practice* **1**, **1**, 3–17.

Hendricks, E., Engler, D., and Fam, M. (2000). *A Generic Mean Value Engine Model for Spark Ignition Engines.* Institute of Automation. Technical Report. Denmark Technical University. www.iau.dtu.dk/~eh/.

Jakubek, S., and Strasser, T. (2002). Fault diagnosis using neural networks with ellipsoidal basis functions. *Proc. American Control Conf.*, May 8-10, Anchorage, AK, 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*, **13**, 189–203.

Lee, S. J., Park, K., Hwang, T. H., Hwang, J. H., Jung, Y. C., and Kim, Y. J. (2007). Development of hardware-in-the-loop simulation system as a test bench for ESP unit. *Int. J. Automotive Technology* **8**, **2**, 203–209.

Ljung, L. (1999). *System Identification - Theory for the User*, 2nd edn. Prentice-Hall. 361–369.

Manzie, C., Palaniswami, M., and Watson, H. (2001). Gaussian networks for fuel injection control. *Proc. IMechE Part D, Automobile Engineering*, **215**, 1053–1068.

Nyberg, M., and Stutte, T. (2004). Model based diagnosis of the air path of an automotive diesel engine. *Control Engineering Practice,* **12**, 513–525.

Official Journal of the European Communities L 350/1 (1998). Directive 98/69/EC of the European Parliament and of the Council of 13 October 1998 relating to measures to be taken against air pollution by emissions from motor vehicles and amending Council Directive 70/220/EEC.

Reineman, M. (2000). *Effectiveness of OBD II Evaporative Emission Monitors – 30 Vehicle Study.* U.S. Environmental Protection Agency Report. October. EPA420-R-00-018.

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. IMechE, Part D, J. Automobile Engineering*, **220**, 1641–1655.

Shayler, P. J., Goodman, M., and Ma, T. (2000). The exploitation of neural networks in automotive engine management systems. *Engineering Applications of Artificial Intelligence* **13**, **2**, 147–157.

Tan, Y., and Saif, M. (2000). Neural-networks-based nonlinear dynamic modelling for automotive engines. *Neurocomputing*, **30**, 129–142.

Wu, J.-D., and Chen, J.-C. (2006). Continuous wavelet transform technique for fault signal diagnosis of internal combustion engines. *NDT & E Int.* **39**, **4**, 304–311.

Yu, D. L., and Gomm, J. B. (2003). Implementation of neural network predictive control to multivariable chemical reactor. *Control Engineering Practice* **11**, **11**, 1315–1323.

Zhang, X. D., Polycarpou, M. M., and Parisini, T. (2002). A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. *IEEE Trans. Automatic Control* **47**, **4**, 576–590.