International Journal of Automation and Computing 05(2), April 2008, 109-118
DOI: 10.1007/s11633-008-0109-9

Robustness Assessment and Adaptive FDI for Car Engine

Mahavir Singh Sangha Dingli Yu* J. Barry Gomm
Control Systems Research Group, School of Engineering, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK

Abstract: A new on-line fault detection and isolation (FDI) scheme proposed for engines using an adaptive neural network classifier is evaluated for a wide range of operational modes to check the robustness of the scheme in this paper. The neural classifier is adaptive to cope with the significant parameter uncertainty, disturbances, and environment changes. The developed scheme is capable of diagnosing faults in 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 these changes occurring at the same time. 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.

Keywords: On-board fault diagnosis, automotive engines, adaptive neural networks (ANNs), fault classification, robustness assessment.

1 Introduction

An automotive engine is a complex machine which is controlled and monitored by a sophisticated electronic system called an 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 since 2003, must have on-board diagnostic systems to monitor engine emissions. These systems were introduced in line with European Directive 98/69/EC[1] 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 1996.

Some engine faults can lead to increase in emissions and affect fuel efficiency adversely. Some serious faults can even lead to ceasing of the engine or even an accident and that is why fault detection, isolation, and accommodation became 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[2]. 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 researchers[3−6]. Earlier work on fault diagnosis of an automotive engine based on parity equations derived from an engine model was presented in [7]. The application of 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 [8]. 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[9]. 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 in [10] for fault diagnosis of six different faults in starter motors which made it possible to diagnose the faults before they occur by keeping fault records of past occurrences. The effectiveness of a non-linear principal component analysis based (PCA-based) monitoring scheme was illustrated for drifting fault in the fuel flow sensor due to a partial blockage of the intercooler in a Volkswagen TDI 1.9 litre diesel engine in [11]. 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 in [12]. A continuous wavelet transform technique for the fault signal diagnosis in an internal combustion (IC) engine and its cooling system was presented in [13]. A neural network model-based fault classification system for a nonlinear dynamic process was investigated in [14] and the real data experiment showed that sensor faults could be detected and isolated even without a process mathematical model. An fault detection and isolation (FDI) scheme for abrupt and incipient faults presented in [15] using online estimators is a good example of an automated fault-diagnosis methodology.

In this paper, two component and two sensor faults with four different levels of intensities have been investigated as four typical and practical examples of spark ignition (SI) engine faults. The faults considered are realistic and have been considered by previous authors in [16, 17]. The two

Manuscript received September 27, 2007; revised January 10, 2008
This work was supported by Universities UK, Faculty of Technology and Environment and School of Engineering, Liverpool John Moores University, UK.
*Corresponding author. E-mail address: d.yu@ljmu.ac.uk

component faults are exhaust gas recycle (EGR) valve stuck up and gas leakage in the intake manifold. The two sensor faults are intake manifold pressure and temperature sensor faults.

A new on-line FDI scheme proposed for engines using an adaptive neural network classifier in [18] is thoroughly tested for a wide range of operational modes to check the robustness of the proposed scheme in this paper. 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 environment 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 a road, i. e., change in speed set-point, load, and engine parameter. The nobility of this paper consists in the successful demonstration of robustness of the developed adaptive neural-network-based FDI algorithm.

Notations

t : Time (s).
α : Throttle plate angle (degree).
n : Engine speed (rpm/1000).
$\dot{m}_f$: Engine port fuel mass flow (kg/s).
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/s).
$\dot{m}_{ap}$: Air mass flow into intake port (kg/s).
T_{EGR} : EGR temperature (Kelvin).
$\dot{m}_{EGR}$: EGR mass flow (kg/s).
V_i : Manifold + port passage volume (m^3).
R : Gasconstant = 287×10^{-5} kJ/kg/Kelvin.
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 ($\lambda = 1$ corresponds to air/fuel ratio of 14.7).
P_p : Pumping power (kW).
P_p : Fuel lower heating valve (kJ/kg)
$\Delta\tau_d$: Injection torque delay time (s).

2 Fault diagnosis method

According to the engine air path dynamics, four variables are chosen as the network inputs: the throttle angle, the 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 with 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 Fig. 1.

The feedback system in a real automotive is a human element (driver). The speed is tried 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 the four input and outputs are conditioned and 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. 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[19]. 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:

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

where $x(k)$ is the network input vector at iteration k, c_i is

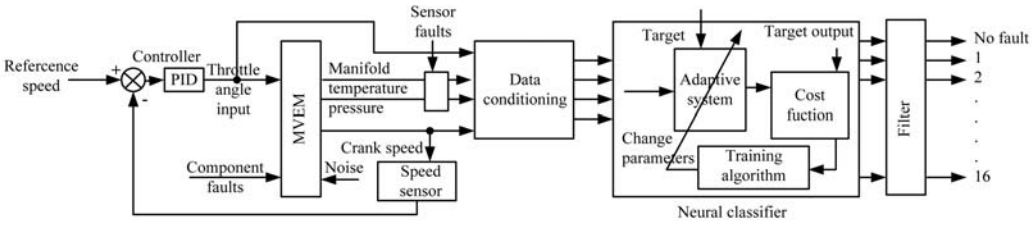


Fig. 1 Information flow of the fault diagnosis

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 and $0 < \alpha < 1$. The complete mathematical derivation of the above equation can be found in [18].

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 is 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 in [20].

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

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

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 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.

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

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 Fig. 2.

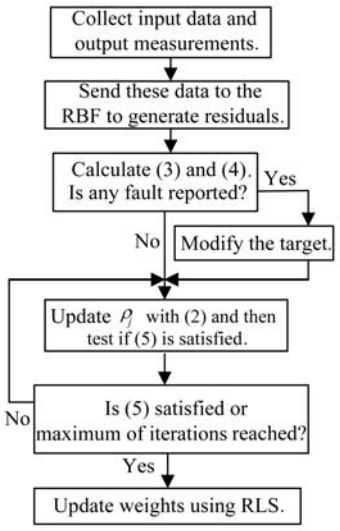


Fig. 2 Flow chart of fault diagnosis and classifier updating

3 Engine dynamics and controller design

3.1 Mean value engine model

An 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 Fig. 3. In a real automotive, the speed feedback control and accordingly the manipulated variable, throttle angle, is given by a human element (driver). But here, 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 Fig. 4. Its dynamics are briefly presented as follows, and the physical parameters are defined in the Nomenclature. The details of the dynamics can be referred in [21].

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

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

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 brackets show the in-flow of gas and the negative term shows the outflow of gas from the intake

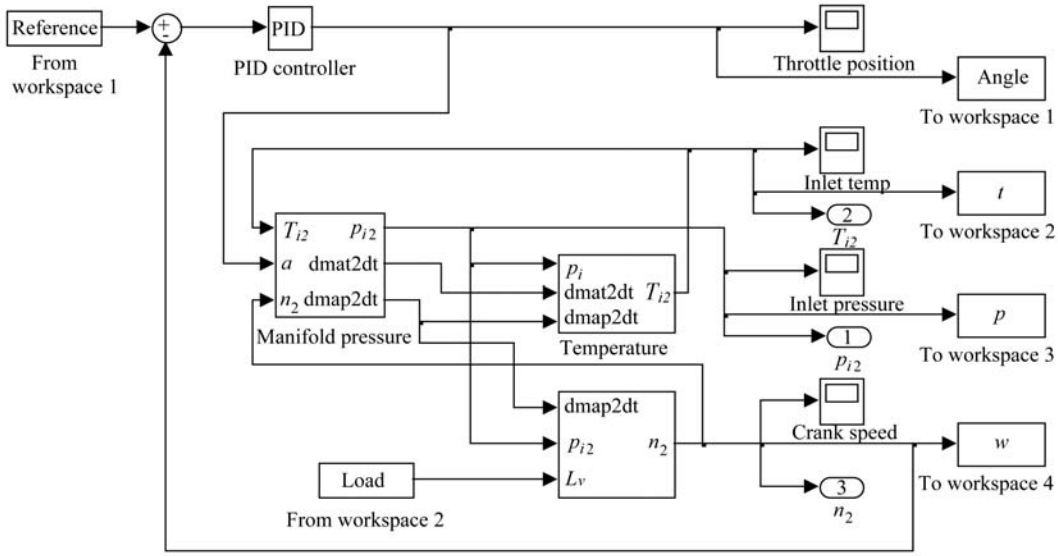


Fig. 3 Simulink model of MVEM with crankshaft speed feedback

manifold (see Fig. 4). Using the law of energy conservation, a state equation which describes the time development of the intake manifold temperature can be given as

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

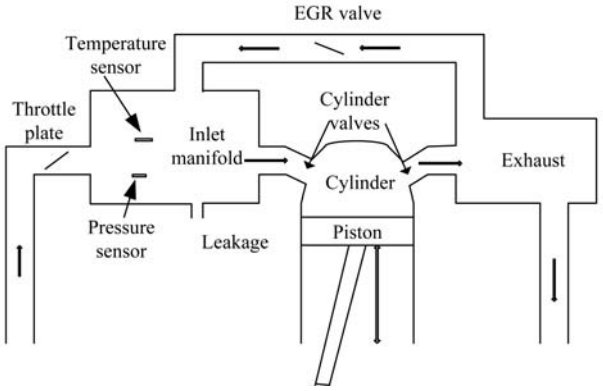


Fig. 4 Schematic of the air intake and exhaust system of the engine

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 (8).

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

where I is the scaled moment of inertia of the engine and its load and the mean injection/torque time delay has been taken into account with variable $\Delta\tau_d$. $\lambda = 1$ corresponds to air/fuel ratio (AFR) of 14.7 for gasoline and 14.5 for diesel. At $\lambda = 1$, 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 Fig. 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,

$$u(t) = K_p e(t) + K_i \int e(t)\mathrm{d}t + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}.$$

This signal u is input to 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 derivative gain. Therefore, the derivative gain is kept zero to keep the controller as simple as possible.

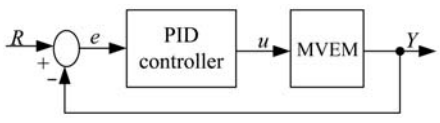


Fig. 5 Closed-loop PID controller

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 s, and therefore 12 data points are collected in every 6 s. The output response of the crankshaft speed for no fault case for five different reference signals is shown in Fig. 6. The output crankshaft speed follows the input reference speed without much overshoot, delay time, and zero steady-state error. The chosen PID ($K_p = 10$, $K_i = 10$, and $K_d = 0$) settings give an acceptable level of performance of the controller for further experimentation.

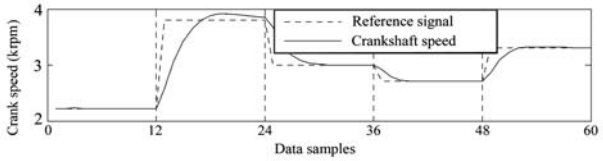


Fig. 6 No fault outputs for five random speed reference signals

4 Fault diagnosis

4.1 Simulating faults

The sensor faults can occur due to two reasons:

1) Ageing and wear & tear of the mechanical parts of the deflection meter;

2) 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 ageing and mechanical faults cause an incorrect meter reading, i. e., over-reading or under-reading of the actual values. Both cases of under-reading 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:

1) Missing gas caps;

2) Loose gas caps;

3) Leaks in gas caps or vapour vent lines;

4) Disconnected purge lines[22].

Current OBD regulations require monitoring of any leaks (for 2003 year model and after) that exceed 0.02 inch in diameter (0.5 mm approximately). It is not practical to create some component faults in a running engine in real life, such as air leakage in the manifold or a stuck EGR valve. 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 the failure of the EGR valve positioning system, which have not been investigated in this paper. The investigation is focused on the detection and isolation of the fault and its intensity, not on pin-pointing the actual component failure of the EGR system:

1) EGR open circuit fault,

2) EGR vent solenoid fault,

3) EGR step motor 1 fault,

4) EGR step motor 2 fault,

5) EGR vacuum regulator fault,

6) EGR boost solenoid control fault etc.

Details of the simulation of the faults are described next.

No Fault: For no fault situation, 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.

Air Leakage Fault: To collect the engine data subjected to the air leakage fault, (6) is modified

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

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.

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 EGR valve stuck up completely and 100% corresponds to full EGR air mass flow, i. e., no fault condition.

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

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

Number	Fault names	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	Temperature sensor 20% over reading	MF=1.2
11	Temperature sensor 10% over reading	MF=1.1
12	Temperature sensor 10% under reading	MF=0.9
13	Temperature 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

4.2 Network training

Two RBF networks are used for fault classification, with one for non-adaptive classifier and the other for adaptive classifier. Both networks have the same structure and will be trained with the same training data and 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 Fig. 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) 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 one fault or no-fault condition. Twenty data sets for different initial and final throttle angle positions are collected as shown in Table 2.

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

Start degree of θ	Accelerating	Decelerating	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

The data sets which are reserved for testing are not used for training. Many data sets were used for testing but at-a-time only two sets were used. For example when two data sets 26–34 and 34–26 are reserved for testing, then the remaining 18 data sets are used for testing. Later on another two sets are reserved for testing ($22 \rightarrow 34$ and $34 \rightarrow 26$) and remaining sets are used for training and so on. As each training data set has the same pattern for 17 faults, one training target matrix X_o (see Fig. 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 and the other entries are zeros, the second column has ones from the 13th row to the 24th row and the other entries are zeros, the last column has ones from the 193rd row to the 204th row and the other entries are zeros.

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 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, and 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.

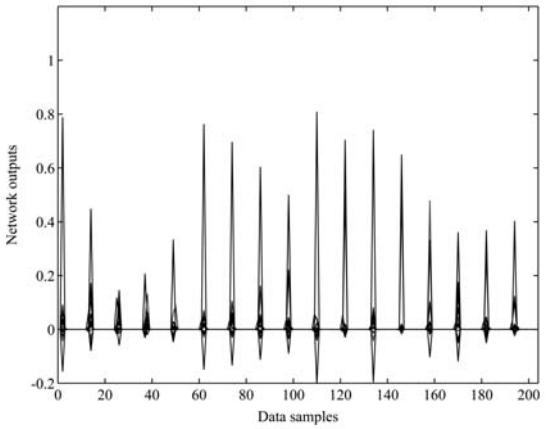
$$\begin{array}{cc} \text{Row numbers} & X_o \\ 1 \sim 12 & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & \cdots & & & \\ 0 & \vdots & \ddots & & & \\ 0 & & & \ddots & & \\ 0 & & & & 1 & \\ \vdots & & & & & \ddots \end{bmatrix} \\ 13 \sim 24 & \\ \vdots & \\ 193 \sim 204 & \end{array}$$

Fig. 7 Target matrix X_o

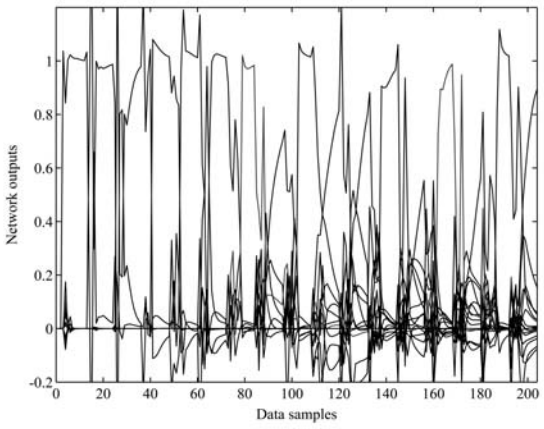
4.3 Fault classification

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

Three different reference signals 2.5 krpm, 3.0 krpm and 3.5 krpm are chosen as Ref 1, Ref 2, and Ref 3 for the speed control, respectively. No fault and faulty data is collected for all the three reference signals, and then both the non-adaptive and the adaptive RBF neural network classifiers are trained and tested for six different sets of data. The results for training the networks on Ref 1 and testing on Ref 3 data are shown in Figs. 8 and 9. The number of centres for the adaptive and non-adaptive networks is chosen as 100.



(a) Non-adaptive classifier (speed fed back and fixed laod)



(b) Adaptive classifier (normal load)

Fig. 8 Networks trained on Ref 1 and tested on Ref 3 (RBF hidden nodes = 100)

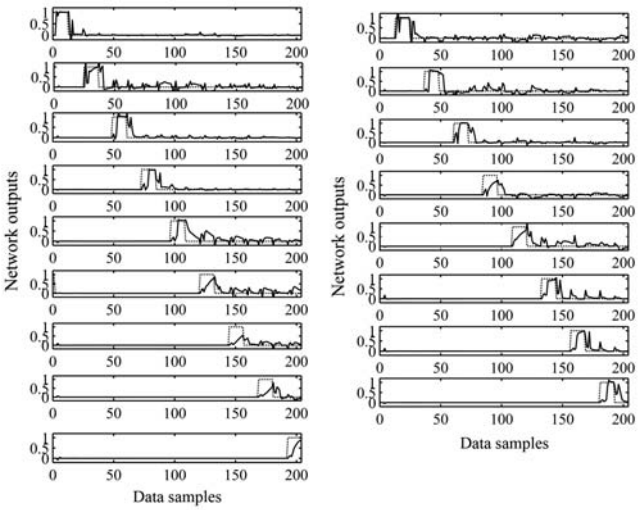


Fig. 9 Details of each fault classification in Fig. 8 (b) shown separately for clarity (Left and right columns show results for state 1, 3, 5, 7, 9, 11, 13, 15, 17, and 2, 4, 6, 8, 10, 12, 14, 16, respectively.)

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. On 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, Fig. 8 (b) is shown in an expanded form in Fig. 9 with every fault classification shown separately. It can be seen that state 1 has one false alarm, state 2 has two false alarms, and 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:

1) Load change;

2) Engine parameter change;

3) All the changes happening simultaneously.

5.1 Load change

To incorporate a provision for engine load change in the MVEM, the crankshaft speed sub-model is modified. The pumping power P_p and 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 (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:

$$\text{Engine load} = K_b \cdot n^3.$$

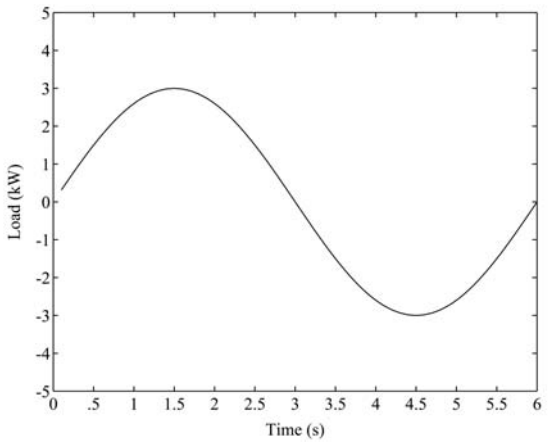
Engine load is equal to load power of engine and therefore,

$$P_b = K_b \cdot n^3.$$

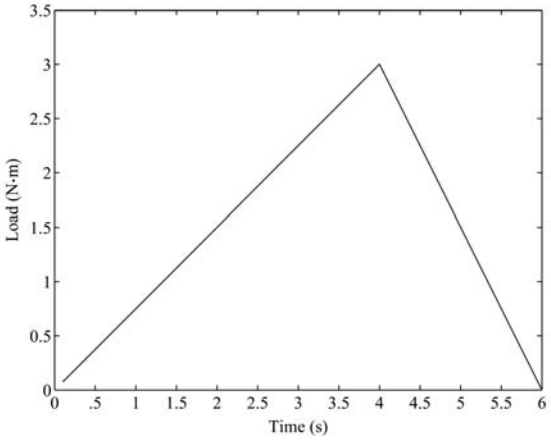
Load power in the modified model is presented as

$$P_b = K_b \cdot n^3 + L_v$$

where L_v is load variation in kW and n is 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 Fig. 10 (a) and (b). The change in load is applied through variable L_v as shown in the simulation diagram in Fig. 2. In the case of a sinusoidal load change, the load on the automobile (engine) can be negative for some time and this represents downhill running of the vehicle. Similarly, increase in load represents uphill running of the vehicle.



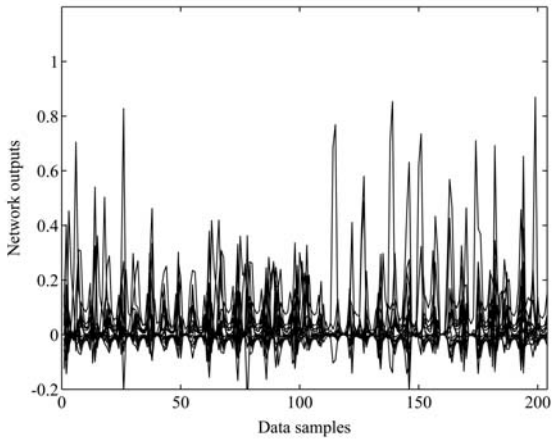
(a) Sinusoidal load change



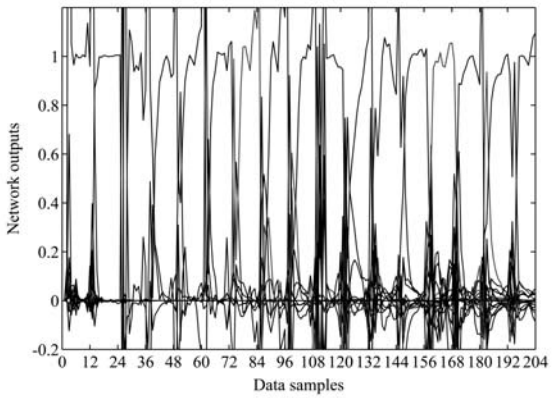
(b) Saw-tooth load change

Fig. 10 Load response of system

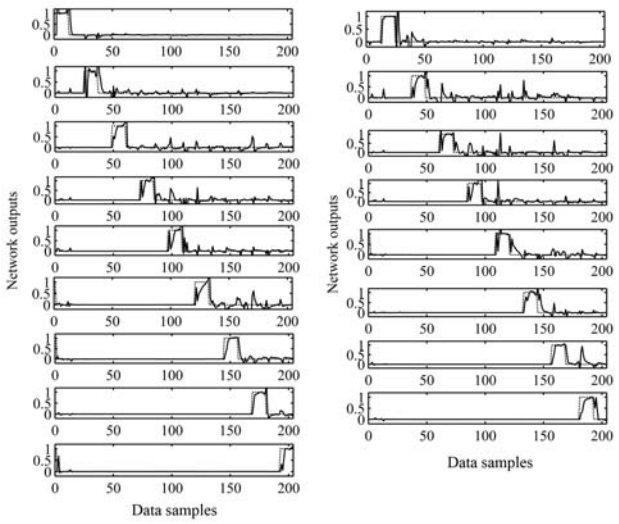
Two sets of data are collected for no fault and faulty conditions; the first set of data for sinusoidal change in load and the second set for saw-tooth change in load. The reference input signal is kept constant at 2.5 krpm for both data sets. First of all, both networks are trained with data for sinusoidal load change and tested with data for saw-tooth load change and then vice-versa. With both training data sets,



(a)



(b)



(c)

Fig. 11 Networks trained on saw-tooth load and tested on sinusoidal load. (a) Results for non-adaptive classifier (Hidden nodes = 90); (b) Results for adaptive classifier (Hidden nodes = 90); (c) Details of each fault classification in (b) shown separately for clarity (Left and right columns show results for state 1, 3, 5, 7, 9, 11, 13, 15, 17, and 2, 4, 6, 8, 10, 12, 14, 16, respectively.)

the classification results were found satisfactory. The classification test results for both classifiers when tested for sinusoidal load change are shown in Fig. 11. The results of the non-adaptive classifier are not good in Fig. 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 Fig. 11 (b) and (c) but with false alarms. There are several false alarms in all and they can be seen in Fig. 11 (c) as small spikes exceeding a threshold of 0.5.

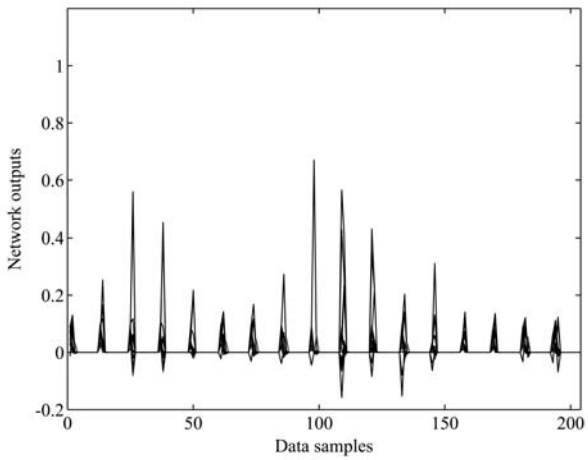
5.2 Engine parameter change

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 ageing 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 the classifiers are trained for the normal engine data and then tested on the data from 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 is much better than the non-adaptive classifier but with false alarms. There are several false alarms as small spikes exceeding threshold of 0.5.

5.3 All the changes happening together

Fig. 12 shows the networks trained on fixed reference and tested on saw-tooth reference, sinusoidal load change and 1% increased engine displacement.

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 spikes in the resultant data. A Butterworth digital filter 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.



(a)

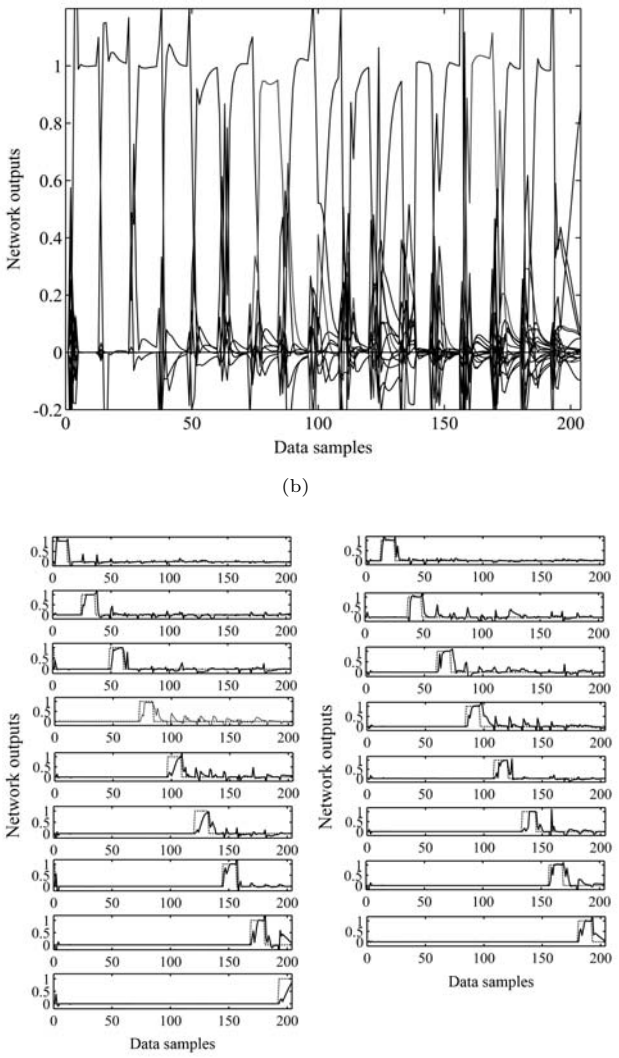


Fig. 12 Networks trained on fixed reference and tested on sawtooth reference, sinusoidal load change and 1% increased engine displacement. (a) Non-adaptive classifier (Hidden nodes = 90, trained on 2 krpm throttle, no load change, no displacement change); (b) Adaptive classifier (Hidden nodes = 90, trained on 2 krpm reference throttle angle, no load change, no displacement change); (c) Details of each fault classification in b shown separately for clarity (Left and right columns show results for state 1, 3, 5, 7, 9, 11, 13, 15, 17, and 2, 4, 6, 8, 10, 12, 14, 16, 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 $0 < W_n < 1.0$, 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 reduced the spikes to half of their original height (i. e., much below the threshold of 0.5) and caused little time delay. The false alarms are practically reduced to zero times as shown in Fig. 13 (a) and (b).

It can be seen from the graphs in Fig. 13 that the spikes

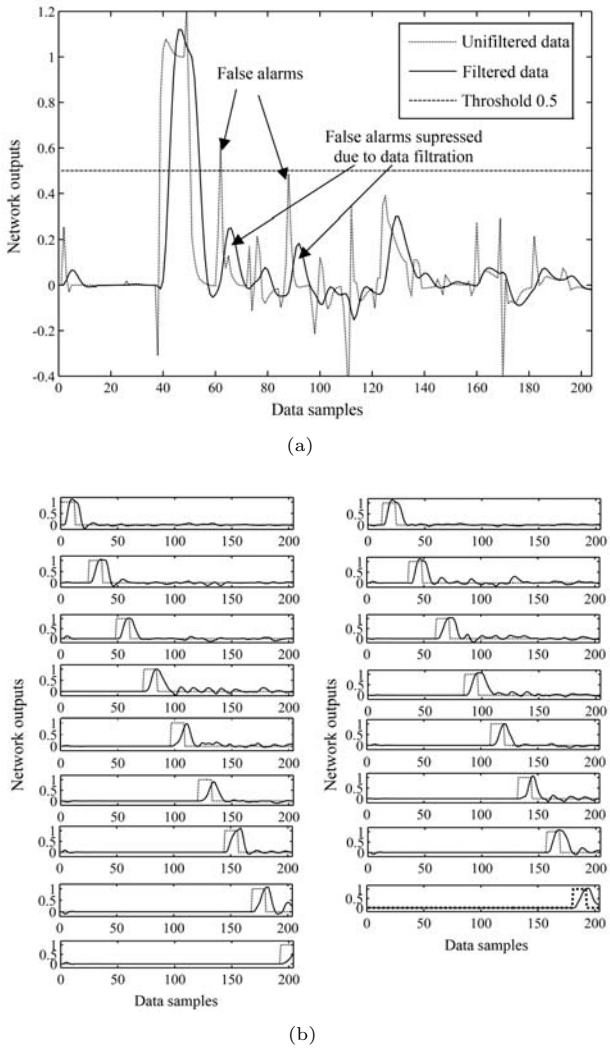


Fig. 13 Classification results after using low-pass Butterworth filter. (a) Comparison of classification result before and after filtration for fault number 4; (b) Classification results of 12 after filtration (Left and right columns show results for state 1, 3, 5, 7, 9, 11, 13, 15, 17, and 2, 4, 6, 8, 10, 12, 14, 16, respectively.)

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 the sixteen faults considered. The non-adaptive classifier fails to cope up with the load change, parameter change, etc. and thus 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 are considered for future work. The changes in four or more engine parameters are also considered for future work.

References

[1] Official Journal of the European Communities L 350/1. 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, 1998.

[2] C. Evans-Pughe. Learning to Drive [Tightening Emissions Regulations]. *Engineering & Technology*, vol. 1, no. 2, pp. 42–45, 2006.

[3] Y. Tan, M. Saif. Neural-networks-based Nonlinear Dynamic Modelling for Automotive Engines. *Neurocomputing*, vol. 30, no. 1, pp. 129–142, 2000.

[4] F. Kimmich, A. Schwarte, R. Isermann. Fault Detection for Modern Diesel Engines Using Signal and Process Model-based Methods. *Control Engineering Practice*, vol. 13, no. 2, pp. 189–203, 2005.

[5] C. Manzie, M. Palaniswami, H. Watson. Gaussian Networks for Fuel Injection Control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 215, no. 10, pp. 1053–1068, 2001.

[6] S. Jakubek, T. Strasser. Fault Diagnosis Using Neural Networks with Ellipsoidal Basis Functions. In *Proceedings of American Control Conference*, IEEE Press, Anchorage, USA, vol. 5, pp. 3846–3851, 2002.

[7] J. J. Gertler. M. Costine, X. W. Fang, R. Hira, Z. Kowalczuk, Q. Luo. Model-based On-board Fault Detection and Diagnosis for Automotive Engines, *Control Engineering Practice*, vol. 1, no. 1, pp. 3–17, 1993.

[8] D. Antory. 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, no. 2, pp. 795–808, 2007.

[9] S. J. Lee, K. Park, T. H. Hwang, J. H. Hwang, Y. C. Jung, Y. J. Kim. Development of Hardware-in-the-loop Simulation System as a Testbench for ESP Unit. *International Journal of Automotive Technology*, vol. 8, no. 2, pp. 203–209, 2007.

[10] O. F. Bay, R. Bayir. Kohonen Network Based Fault Diagnosis and Condition Monitoring of Pre-engaged Starter Motors. *International Journal of Automotive Technology*, vol. 6, no. 4, pp. 341–350, 2005.

[11] D. Antory, U. Kruger, G. Irwin, G. McCullough. Fault Diagnosis in Internal Combustion Engines Using Non-linear Multivariate Statistics. *Proceedings of the Institution of Mechanical Engineers – Part I: Journal of Systems and Control Engineering*, vol. 219, no. 14, pp. 243–258, 2005.

[12] P. J. Shayler, M. Goodman, T. Ma. The Exploitation of Neural Networks in Automotive Engine Management Systems. *Engineering Applications of Artificial Intelligence*, vol. 13, no. 2, pp. 147–157, 2000.

[13] J. D. Wu, J. C. Chen. Continuous Wavelet Transform Technique for Fault Signal Diagnosis of Internal Combustion Engines. *NDT & E International*, vol. 39, no. 4, pp. 304–311, 2006.

[14] D. L. Yu, J. B. Gomm. Implementation of Neural Network Predictive Control to Multivariable Chemical Reactor. *Control Engineering Practice*, vol. 11, no. 11, pp. 1315–1323, 2003.

[15] X. D. Zhang, M. M. Polycarpou, T. Parisini. A Robust Detection and Isolation Scheme for Abrupt and Incipient Faults in Nonlinear Systems. *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 576–590, 2002.

[16] M. Nyberg, T. Stutte. Model Based Diagnosis of the Air Path of an Automotive Diesel Engine. *Control Engineering Practice*, vol. 12, no. 5, pp. 513–525, 2004.

[17] D. Capriglione, C. Liguori, C. Pianese, A. Pietrosanto. On Line Sensor Fault Detection, Isolation and Accommodation in Automotive Engines. *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 4, pp. 1182–1189, 2003.

[18] M. S. Sangha, D. L. Yu, J. B. Gomm. On-board Monitoring and Diagnosis for Spark Ignition Engine Air Path via Adaptive Neural Networks. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 220, no. 11, pp. 1641–1655, 2006.

[19] J. Moody, C. Darken. Fast Learning in Networks of Locally-tuned Processing Units, *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[20] L. Ljung. *System Identification: Theory for the User*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, USA, pp. 361–369, 1999.

[21] E. Hendricks, D. Engler, M. Fam. A Generic Mean Value Engine Model for Spark Ignition Engines, Technical Report, Institute of Automation, Denmark Technical University, [Online], Avaiable: http://www.iau.dtu.dk/~eh/, 2000.

[22] M. Reineman. Effectiveness of OBD II Evaporative Emission Monitors – 30 Vehicle Study, U.S. Environmental Protection Agency Report EPA420-R-00-018, 2000.

Mahavir Singh Sangha received first class B. Eng. (Electrical) degree form University of Jodhpur, India, in 1991 and M. Sc. (Intelligent Control) degree with distinction from Liverpool John Moores University (LJMU), UK, in 2005. He is presently a Ph. D. candidate at LJMU, Liverpool, UK. He worked in an electricity distribution company for about 12 years.

His research interests include fault diagnosis in an automotive engine and neural networks.

Dingli Yu received the B. Eng. degree from Harbin University of Civil Engineering, China, in 1982, the master degree from Jilin University of Technology (JUT), China, in 1986, and the Ph. D. degree from Coventry University, UK, in 1995, all in electrical engineering. He was a lecturer at JUT from 1986 to 1990 before he came to University of Salford as a visiting researcher in 1991. He then worked at Liverpool John Moores University as a post-doctoral researcher since 1995 and became a lecturer in 1998, where he is currently a professor of control systems.

His research interests include fault detection and fault tolerant control of bilinear and nonlinear systems, adaptive neural networks and their control applications, and model predictive control for chemical processes and engine systems.

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 modelling, control and fault diagnosis of non-linear processes, intelligent methods for control, system identification, adaptive systems, chemical process, and automotive applications.