Sensor fault detection, isolation, accommodation and unknown fault detection in automotive engine using AI

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

1 Cummins Engine Co. Ltd, Royal Oak Way South, Daventry, NN11 8NU, UK
2 Control Systems Research Group, School of Engineering, Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK
*Corresponding author: Email d.yu@ljmu.ac.uk

Abstract

Sensor fault detection, isolation (FDI) and accommodation has been investigated along with detection of unknown faults for an automotive engine. Radial basis function (RBF) neural networks are used for fault diagnosis. The RBF network is trained off line with K-means and batch least squares (BLS) algorithms. No fault and fault data are simulated in Matlab for four different sensors e.g. throttle angle position, crankshaft speed, and inlet manifold pressure and temperature sensors. All the sensors are investigated for ten percent positive and negative bias faults and also for unknown faults. Simulations show satisfactory results for FDI. Further, the fault accommodation for three sensors is also investigated using predictive neural networks and the results with acceptable levels of errors are achieved.

Keywords: fault detection, training, radial basis function, simulation, fault isolation

DOI: http://dx.doi.org/10.4314/ijest.v4i3.4

1. Introduction

Nowadays a large number of sensors are used in a car for improved reliability and increased comfort. From seat adjusting to reverse parking, sensors play a vital role in a modern car. Mercedes ‘S’ class has 40 microprocessors and over 100 motors (Denton, 2004) for comfort, control and FDI. Sensors used by motor vehicle systems are following a trend towards greater integration of processing power in the actual sensor. Integrated pre processing unit and analogue to digital converter in a sensor, make the signal interference proof. Sensors with local intelligence, known as ‘intelligent sensors’, are also available commercially but are very expensive. This level of integration allows built in monitoring and diagnostic ability in a sensor which leads to much improved reliability and control of the vehicle. The correctness of operation of an electronic control unit (ECU) of car engine depends upon a number of measurements e.g. crankshaft speed, inlet manifold pressure, throttle position, air fuel ratio etc. A growing demand on security and comfort has pushed an increased use of suitable sensors and actuators. In-vehicle conditions are optimised using environmental sensors for evaluating the outside conditions e.g. road condition, visibility, presence of ice and intensity of rain etc. Anti-crash sensors, GPS, antiskid breaking and anti-spin traction control etc. are used in most of the new cars for improved security and comfort. Most importantly, to avoid crashes due to human errors, the driver status/condition is also monitored on-line by a more complex measurement system in high end cars. Automotive measurement system should also be sensor fault tolerant like aircrafts and nuclear power plants.

Sensor fault correction is essentially a three fold system. The first stage is fault detection, which indicates occurrence of a known or unknown fault. The second stage involves establishment of type and location of the fault and is called fault isolation. When a fault is detected and isolated, it may be desirable to auto correct the fault if possible. The auto correction of fault may not be
possible for component and unknown faults; but known sensor faults can be corrected according to their known no-fault characteristics. This is known as sensor fault accommodation.

In this paper four different sensors have been investigated for fault diagnosis e.g. crankshaft speed, throttle angle position, inlet manifold temperature and pressure sensors. A mean value engine model (MVEM) is used for experimentation. The MVEM is run in Matlab/Simulink and no fault data is collected for all the sensors under consideration. 10% positive and negative bias faults are simulated on all the four sensors for different throttle angle inputs to the engine. These faults considered are realistic and have been considered by previous authors in Capriglione et al. (2003) as well as Nyberg and Stutte (2004).

Good classification capability of RBF neural networks are exploited for fault detection and isolation (FDI). K-means algorithm is well known for RBF network training. K-means in conjunction with batch least squares (BLS) algorithm is used to calculate appropriate values for RBF network training i.e. centres for hidden nodes, width parameter and weights for the output layer. The trained network is then tested for different sets of data and acceptable FDI results are achieved. Further, unknown fault detection is also demonstrated by simulation results for different known faults. When none of the ‘no fault’ or ‘fault’ output neurons exceed the threshold, it can be interpreted as detection of ‘unknown fault’ (Li et al., 2002). This process is also known as novelty detection (Bishop, 1994). Any fault in the system for which the RBF classifier is not trained is categorised as unknown fault. The RBF classifier is only trained for the sensor faults and thus any component fault happened in the system would be treated as unknown fault. For instance if there is an air leakage in the inlet manifold or the EGR valve is stuck up in closed position. There can be other unknown faults which may change some vital parameter like the air/fuel ratio. Lastly, the sensor fault accommodation is achieved for three different sensors using predictive neural networks as per their known no-fault characteristics.

2. Engine Dynamics and Faults

2.1 Engine Dynamics

It is impracticable to introduce faults into a running automotive engine and therefore to investigate the feasibility of the adaptive neural network model based fault diagnosis system for SI engines, engine simulations are used instead of a real engine test bed. The engine simulation model used here is based on the MVEM (Hendricks et al., 2000), a well-known and widely used benchmark for engine modelling, control and fault diagnosis. It consists of three sub-models that describe the intake manifold dynamics including air mass flow, pressure and temperature and the crankshaft speed as illustrated in Fig.1

Fig 1: the MVEM Simulation diagram
2.2 Manifold Filling Dynamics

The manifold filling dynamics in reality is based on an adiabatic operation rather than isothermal. The manifold pressure can be represented by equation (1).

\[
\frac{dP_i}{dt} = \frac{\kappa R}{V_i} \left( -m_{ap} T_i + \dot{m}_{at} T_a + \dot{m}_{EGR} T_{EGR} \right)
\]

(1)

The positive terms within brackets show the in-flow of gas and the negative term shows the outflow of gas from the intake manifold. Using the law of energy conservation a state equation which describes the time development of the intake manifold temperature can be given as:

\[
\frac{d\dot{T}_i}{dt} = \frac{RT_i}{P_i V_i} \left[ -m_{ap} (\kappa - 1) \dot{T}_i + \dot{m}_{at} (\kappa T_a - T_i) + \dot{m}_{EGR} (\kappa T_{EGR} - T_i) \right]
\]

(2)

2.3 Crank Shaft Speed Dynamics

Applying the law of conservation of rotational energy, the crankshaft dynamics of an SI engine MVEM is described by equation (3).

\[
\dot{n} = -\frac{1}{I_n} \left( P_f(p_1, n) + P_p(p_1, n) + P_h(n) \right) + \frac{1}{I_n} \eta_i(p_1, n, \lambda) \dot{m}_f(t - \Delta \tau_d)
\]

(3)

where \( I_n \) 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 \( \Delta \tau_d \).

2.4 Fault Simulation

In this paper four sensor faults have been investigated as four typical and practical examples. All the four faults are considered with positive and negative bias of 10%. The sensor faults can occur due to two basic reasons: (i) wear and tear of the mechanical parts of the deflection meter or some change in the value of resistance or capacitance used in the circuit due to aging (e.g. leakage of dielectric material or change in dielectric strength of the material), and (ii) electrical fault such as short circuit or open circuit 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 and can be detected by a value check on the sensed value of the variable under measurement. On the contrary the aging and mechanical faults cause an incorrect meter reading and thus are difficult to detect. They can cause to over read or under read the sensed variable and are known as bias faults.

A.) 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 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 no component is malfunctioning. The no fault data is collected for different throttle angle inputs ranging from 20 to 40 degree (the idling throttle angle for the engine is 15 degrees) for different operating points.

B.) Sensor Faults: Different multiplying factors (MFs) are used to generate fault data for eight different sensor faults as shown in table 1.
Table 1: 8 fault and no fault states and multiplying factors

<table>
<thead>
<tr>
<th>State</th>
<th>Fault</th>
<th>M.F.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Fault (NF)</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>Throttle angle position sensor 10% under reading</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>Throttle angle position sensor 10% over reading</td>
<td>1.1</td>
</tr>
<tr>
<td>3</td>
<td>Intake Manifold pressure sensor 10% under reading</td>
<td>0.9</td>
</tr>
<tr>
<td>4</td>
<td>Intake Manifold pressure sensor 10% over reading</td>
<td>1.1</td>
</tr>
<tr>
<td>5</td>
<td>Intake Manifold temperature sensor 10% under reading</td>
<td>0.9</td>
</tr>
<tr>
<td>6</td>
<td>Intake Manifold temperature sensor 10% over reading</td>
<td>1.1</td>
</tr>
<tr>
<td>7</td>
<td>Crankshaft Speed sensor 10% under reading</td>
<td>0.9</td>
</tr>
<tr>
<td>8</td>
<td>Crankshaft Speed sensor 10% over reading</td>
<td>1.1</td>
</tr>
</tbody>
</table>

C.) Unknown Faults: Three different unknown faults are simulated in the model:
(i) Air leakage in the intake manifold
(ii) EGR valve clogged in closed position
(iii) Air/fuel ratio
   (a) Thin Air/fuel ratio (24.7)
   (b) Thick Air/fuel ratio (10.7)

(i) Air leakage in the intake manifold

Equation (1) can be modified as

\[ \hat{P}_i = \frac{kR}{V_i} \left( -\dot{m}_{ap} T_i + \dot{m}_{at} T_a + \dot{m}_{EGR} T_{EGR} - k \right) \]  

where constant \( k \) is added in the model to change the outflow of the intake manifold, and this increase in the outlet is treated as air leakage. The air leakage is decided as a percentage of the total air intake in the intake manifold.

(ii) EGR valve clogged in closed position

The normal value of EGR is kept as 1/6 of the total air mass flow, i.e. 16.67%. The EGR can be as high as 20% of the total air mass flow in the intake manifold. Thus, a realistic value of EGR feedback is chosen for the experiments. The value of \( \dot{m}_{EGR} \) is taken as 0% of the normal EGR air mass flow which corresponds to the EGR valve clogged in fully closed position.

(iii) Air/fuel ration fault

For a normal operation, \( \lambda \) is taken equal to “1” which corresponds to air/fuel ratio of 14.7 for gasoline and 14.5 for diesel. At \( \lambda \) equal to "1", it is stoiciometry or the point at which the most complete combustion takes place. \( \lambda \) gives a measure of Air Fuel Ratio which is independent of the type of fuel being used. \( \lambda \) more than one implies excess air and less than one implies excess fuel. \( \lambda > 1.0 \Rightarrow \text{Excess Air (Lean)} \) and \( \lambda < 1.0 \Rightarrow \text{Excess Fuel (Rich)} \). The air/fuel ratio is forced to 24.7 for excess air fault and is forced to 10.7 for excess fuel fault.

3. RBF Training Algorithms

Training an RBF network implies optimising the parameters of centres, widths and weights in the output layer. For off-line training, the K-means, the P-nearest neighbours and the batch least squares (BLS) algorithms are used. These algorithms are reviewed or derived below.
3.1 K-means Algorithm

The centres are set by the K-means clustering method whose objective is to minimise the sum squared distances from each input data to its closest centre so that the data is adequately covered by the activation functions \( \phi(t) \). The K-means clustering method proceeds as follows:

1. Randomly choose some input data to be the initial centres. The number of the centres is equal to the number of hidden nodes \( n_h \).
2. Let \( k(x) \) denote the index of the best-matching centre for the input vector \( x \). Find \( k(x) \) at iteration \( t \) by minimising the sum squared distances:

\[
\sigma_k = \arg \min \left[ \frac{1}{2} \sum_{k=1}^{n_h} (x(t) - c_k(t))^2 \right]
\]

where \( c_k(t) \) is the centre of the \( k^{th} \) activation function at iteration \( t \).

3. Update the centres of the activation functions by using the following rule:

\[
c_k(t+1) = \begin{cases} 
    c_k(t) + \alpha_c [x(t) - c_k(t)] & \text{if } k = k(x) \\
    c_k(t), & \text{otherwise}
\end{cases}
\]

where \( \alpha_c \) is the centre learning rate that lies in the range \( (0, 1) \).

4. Increment \( t \) by 1 and go back to step 2. Continue the algorithm until no noticeable changes are observed in the centres \( c_k \).

3.2 P-nearest Neighbours Method

The widths are computed by the p-nearest neighbours’ method. The excitation of each node should overlap with other nodes (usually closest) so that a smooth interpolation surface between nodes is obtained. In this method, the widths for each hidden node are set as the average distance from the centre to the \( p \) nearest centres as given by:

\[
\sigma_i = \frac{1}{p} \sum_{d=1}^{p} |c_d(t) - c_i(t)|, \quad i = 1, \ldots, n_h
\]

The value of \( p \) is chosen as 3 for this experimentation but it can be different for other problems.

3.3 Batch Least Square (BLS) Algorithm

This algorithm is widely used for off-line training. If the RBF network has \( d \) inputs, \( q \) outputs and \( n_h \) hidden nodes, the output matrix with \( N \) samples \( (\hat{Y}_{N \times q}) \) can be written as

\[
\hat{Y} = \Phi(X)W
\]
Where \( X^{Nxt} \) is the input matrix, \( \Phi(X)^{Nxt, nh} \) is the matrix of activation function outputs and \( W^{nh, eq} \) is the matrix of weights. If there is a modelling error \( E \), then the modelling target is \( Y = \hat{Y} + E \). Suitable weights are calculated in order to minimise the modelling error. Weights \( W \) for the BLS training algorithm are given as:

\[
W = (\Phi^T \Phi)^{-1} \Phi^T Y
\]  

(9)

This training algorithm can update the weights with a batch of training data. The optimal weights will be fixed and used for modelling once they have been determined for minimum modelling error.

### 3.4 Training & Testing Procedures

Procedure for the training algorithm is explained step by step as follows:

a) K-means and \( P \)-nearest neighbours algorithms are used to get the widths and the centres in the hidden layer of an RBF network utilising the first set of the data collected from simulation.

b) Gaussian basis function is used as activation function and the activation function outputs are calculated.

c) Weights in the output layer are calculated by using batch least squares (BLS) algorithm for minimum modelling error for the pre-defined target values.

d) All the three calculated matrices viz. centres, widths and weights are saved for the testing phase.

e) The second set of data is used for the testing purpose and passed through the activation function. It utilises the previously calculated values of centres and widths. The activation function outputs are calculated and stored in a matrix.

f) The output of RBF is calculated using previous values of the weights and the new outputs of the activation function.

g) The output is compared against the ideal target values and 0.5 is considered as the decision boundary.

h) Entire procedure from a) to g) is repeated with other sets of data.

### 3.5 Data Flow for FDI

The Fig 2 shows the block diagram of the FDI system. First of all the MVEM simulation is run and different data sets for all the four parameters are collected as explained in section 4.1. Then the collected data is normalised as explained in section 4.2. All the
eight faults are simulated on no fault data as explained in section 2.4 before. The neural network is trained on training data set. Then the different testing data sets are passed through the pre-trained neural classifier. The outputs of the classifier are low passed before decision stage. In the decision stage, one of the nine states (from NF to 8) will be high (while others are low) to show the occurrence of no fault or any one of the faults. After the detection of fault, the fault accommodation would take place according to the fault. This is explained in section 4.5 separately.

4. Simulation & Results

4.1 Data Collection

(A.) RBF Training data collection: First of all the MVEM simulation is run for different throttle angle inputs i.e. 20°, 22°, 24°… 40° for no fault condition and data for throttle angle position, inlet manifold pressure, temperature and crankshaft speed is collected. In the same way the data for each fault condition listed in table 1 is collected for all the different throttle angle inputs. 12 data points are collected for each throttle input in 6 seconds at a sample time of 0.5 seconds. 120 data points are collected for each state for 10 different throttle angle inputs. There are 9 states in all and therefore the size of the training data set will be 1080x4 (12x10x9=1080). The target matrix size would be 1080x9 as explained in section 4.2 ahead.

(B.) Testing Data collection for known faults: Several testing data sets are separately collected for five different random throttle angle inputs to the MVEM. Same as before, 12 data points are collected for each throttle input in 6 seconds at a sample time of 0.5 seconds. Therefore, 60 data points are collected for each state for 5 different throttle angle inputs. The size of the testing data set is 540x4 (12x5x9=540).

(C.) Testing Data collection for unknown faults: All the unknown faults are simulated one by one for 6 seconds. Each time when the simulation is run, 60 data points are collected at 0.5 seconds sample time for 5 different throttle angle inputs. Four different data sets are collected; one for each unknown fault i.e. 20% air leakage in the inlet manifold, EGR valve clogged in closed position, air/fuel ration thin (24.7) and air/fuel ratio thick (10.7). The size of each testing data set is 60x4.

4.2 Data Pre-processing

All the data sets collected are pre-processed. One data set is used as training data and the other as test data. The data is normalised by subtracting the steady state values and then scaled to the range of [0 1],

\[ x_{nor} = \frac{x - x_{ss}30}{x_{ss}30} \]

where \( x, x_{ss}, x_{nor} \) are real data, steady state data at 30° throttle and normalized data respectively. By normalisation there will not be any data having much greater numerical value than others and intending to dominate the training.

The target matrix \( xo \) has ones in the first column up to the 120th row and all the other entries are zeros, the second column has ones from the 121st row to the 240th row and so on. The last column has ones from the 961st row to the 1080th row. This is shown as follows:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & \cdots \\
121 & 240 & 0 & 1 & \cdots \\
\vdots & \ddots & 0 & \ddots \\
961 & 1080 & 0 & 1 & \cdots \\
\end{bmatrix}
\]
Thus, one of the 9 columns in the target matrix is associated to a fault condition. With the chosen input variables and the target, RBF network was trained with the training data set, where 90 centres were chosen using the $K$-means clustering algorithm. The widths were chosen using the $p$-nearest algorithm, and the weights were trained using the batch Least Squares (BLS) algorithm.

### 4.3 Detection of known faults

Training and testing procedure given in section 3.4 is followed and the testing results for the second and third sets of data are shown in Fig. 3 and 4.

![Fig. 3: Test results for data set 2](image)

(a) before data filtration  (b) after data filtration

There are a number of visible spikes crossing the threshold of 0.5 in Fig 3a and Fig 4a. These spikes would cause false alarms when actually the fault has not happened. To get rid of these spikes, the data is low passed using a third order Butterworth filter and the results are shown in Fig 3b and Fig 4b. It can be seen that the filtered data has no false alarms and classifies all the faults clearly.

### 4.4 Detection of unknown fault

All the four data sets collected for unknown faults were passed through the pre-trained RBF network and the results are shown in Fig 5, 6, 7 and 8. For the correct classification of an unknown fault, no value of the test result should exceed the threshold of 0.5. The first, third and fourth faults are clearly classified as shown in Fig 5, 7 and 8 but for the second fault (EGR valve clogged in closed position), some test result values exceed the threshold as shown in Fig 6. None of the known fault states and no fault state should exceed the threshold, which did not happen in case of second fault and therefore it was not detected.
Fig. 5: Test results for first unknown fault (20% air leakage in inlet manifold)
(a) Before data filtration and (b) After data filtration

Fig. 6: Test results for second unknown fault (EGR valve clogged in closed position)
(a) Before data filtration and (b) After data filtration
The developed FDI system can only detect and isolate the unknown faults which could change the RBF training parameters to a sufficient level. It implies that if there is an unknown fault which does not affect or just minutely affects the RBF training parameters (i.e. crankshaft speed, throttle angle position, inlet manifold temperature and pressure) will not be detected by the FDI system.

4.5 Fault Accommodation

Three sensors are considered for fault accommodation in this section, e.g. manifold pressure, crankshaft speed and throttle angle position sensors. There are three different artificial neural networks (ANNs) to predict correct value for the faulty sensor as shown in Fig 9. The present value and three past instance values of manifold pressure \([p(k), p(k-1), p(k-2) \text{ and } p(k-3)]\) and crankshaft speed \([n(k), n(k-1), n(k-2) \text{ and } n(k-3)]\) are used to predict the present value of throttle angle position \([th(k)]\) in \(ANN_{th}\). Similarly the present value and three past instance values of throttle angle position \([th(k), th(k-1), th(k-2) \text{ and } th(k-3)]\) and manifold pressure \([p(k), p(k-1), p(k-2) \text{ and } p(k-3)]\) are used to predict the present value of the crankshaft speed \([n(k)]\) and present value and three
past instance values of throttle angle position \([\text{th}(k), \text{th}(k-1), \text{th}(k-2) \text{ and } \text{th}(k-3)]\) and crankshaft speed \([n(k), n(k-1), n(k-2) \text{ and } n(k-3)]\) are used to predict the present value of manifold pressure \([p(k)]\) in \(\text{ANN}_n\) and \(\text{ANN}_p\) respectively.

Only the ANN which corresponds to the detected faulty sensor is started at a given time. Each ANN gives the correct value of the corresponding faulty sensor, starting from the measured values of the other two fault free sensors. The ANN architecture constitutes of a radial basis function network with 20 hidden nodes. Therefore the structure of each artificial neural network (ANN) is \(8\times20\times1\). The same RBF training algorithms are used as for the FDI system before.

![Diagram of three artificial neural networks (ANNs) for predictive values of throttle angle, crankshaft speed and inlet manifold pressure respectively.](image)

**Fig. 9:** Three artificial neural networks (ANNs) for predictive values of throttle angle, crankshaft speed and inlet manifold pressure respectively

![Comparison of actual and predicted values of throttle angle](image)

**Fig. 10:** Comparison of predicted and actual values of throttle angle, crankshaft speed and manifold pressure in (a), (b) and (c) respectively

The fault accommodation results for throttle angle position, crank shaft speed and inlet manifold pressure sensors are shown in Fig. 10.
4.6 Accommodation performance evaluation

The reconstruction capabilities of the three ANNs are investigated for the purpose of accommodation system performance evaluation. Fig. 10 shows a comparison between the actual sensor outputs and the predicted ones. The average reconstruction error for ANN$_{th}$ and ANN$_{n}$ are below 5% whereas for ANN$_{p}$ it is below 3% as shown in Table 2.

<table>
<thead>
<tr>
<th>Artificial Neural Network (ANN)</th>
<th>ANN$_{th}$ (Predictive throttle angle)</th>
<th>ANN$_{n}$ (Predictive crankshaft speed)</th>
<th>ANN$_{p}$ (Predictive manifold pressure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eight ANN Inputs</td>
<td>n(k)…n(k-3); p(k)…p(k-3)</td>
<td>th(k)…th(k-3); p(k)…p(k-3)</td>
<td>th(k)…th(k-3); n(k)…n(k-3)</td>
</tr>
<tr>
<td>Number of Hidden Nodes</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.0433</td>
<td>0.048</td>
<td>0.028</td>
</tr>
<tr>
<td>Absolute error standard deviation</td>
<td>0.067</td>
<td>0.047</td>
<td>0.058</td>
</tr>
<tr>
<td>Average % error</td>
<td>4.33%</td>
<td>4.8%</td>
<td>2.8%</td>
</tr>
<tr>
<td>Peak % error</td>
<td>20.15%</td>
<td>29.91%</td>
<td>18.53%</td>
</tr>
</tbody>
</table>

In all the tests, a peak error for ANN$_{th}$ and ANN$_{p}$ is nearly 20% whereas for ANN$_{n}$ it is as high as nearly 30%. However, these reconstruction accuracies are acceptable in applications such as automotive engine where typical sensor aging can lead to measurement accuracies of up to 30% (Capriglione et al, 2007).

5. Conclusion

This paper described a model based fault diagnosis system for detection, isolation and accommodation of sensor faults in an automotive engine air path. $K$-means algorithm along with batch least squares (BLS) algorithm was used for RBF neural network training and testing. Modified mean value engine model (MVEM) was used to generate no fault data and then sensor faults were simulated. The sensor faults of $\pm$ 10% amplitude were successfully detected and isolated and also unknown faults were detected. It may not be possible for this fault diagnosis scheme to detect all unknown faults especially those which do not affect the considered sensors up to a certain level. Further the known sensor faults were accommodated with acceptable level of accuracy.

Acknowledgement

The first author would like to thank ‘Universities UK’ for ORSAS award which supported this research financially.

References


Biographical notes

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

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

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

Received December 2010
Accepted April 2011
Final acceptance in revised form July 2011