UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Fault Detection and Isolation for Engine under Closed-Loop Control

Adnan Hamad, Dingli Yu, J B Gomm
Control system research group, school of engineering
Liverpool John Moore University
Liverpool, UK
adnanbohliga@yahoo.com

Mahavir S Sangha
Test Technology & Emissions, Cummins Inc
Daventry, UK

Abstract—**Fault detection and isolation (FDI) have become one of the most important aspects of automobile design. Fault detection and isolation for engine open loop system was investigated in many research. In fact, the simulation results obtained from engine open loop system do not reflect the real situation for automotive engine. In the practice the engine works as closed-loop control system. In this paper, a new FDI scheme is developed for automotive engines under closed-loop control system. Test the method using closed-loop system has been done. The method uses an independent radial basis function (RBF) neural network model to model engine dynamics, and the modeling errors are used to form the basis for residual generation. Furthermore, another RBF network is used as a fault classifier to isolate occurred fault from other possible faults in the system. The performance of the developed scheme is assessed using an engine benchmark, the Mean Value Engine Model (MVEM) with Matlab/Simulink. Six faults have been simulated on the MVEM, including four sensor faults, one component fault and one actuator fault. The simulation results show that all the simulated faults can be clearly detected and isolated in dynamic conditions throughout the engine operating range.**

Keywords: Automotive engines under closed-loop control, independent RBF model, RBF neural network, fault detection, fault isolation.

I. Introduction

A fault is any type of malfunction of components that may happen in a system and this fault will degrade the system performance. Fault detection is the program which informs us that something wrong in the system and needs to be repaired. Also, fault isolation is way to determine which fault occurs among the possible faults. To detect faults we usually compare the outputs of the real system which is in this paper the MVEM, and the outputs of a neural network model of the engine. Rolf Isermann has proposed Model-based fault-detection and diagnosis methods for some technical processes [1]. On-line sensor fault detection, isolation, and accommodation in automotive engines had been studied by Domenico Capriglione [2]. Fault detection and isolation for MVEM open loop control system were achieved in previous paper of the authors [3]. However, in the practice the engine does not work as an open loop control system. Automotive engine works under close loop control system with feedback control. Fault detection and isolation using close loop system is quite different comparing with open loop system. The main points are: fault detection and isolation is much more difficult than using open loop system. The reason is the process outputs in close loop system will be fed back and this will affect the sensor faults. Secondly, in MVEM open loop system, Air fuel ratio (AFR) was not controlled at the 14.7, because the feature of feed-back did not use in the open loop control system. In this paper, a new fault detection and isolation method will be implemented by using MVEM when this model is under close loop control system. An independent RBFNN model is used to model a dynamic system using RAS throttle angle as an input. Feed-back (FB) and feed-forward (FF) control methods will be applied to the MVEM. The K-means clustering algorithm is used to choose the centres of RBFFNN. Recursive least squares (RLS) algorithm is used to update for each new sample the parameter matrix W.

II. Control structure for MVEM

Fig. 1 shows the Simulink model of the automatic control loop for the MVEM including feed-forward and feed-back controllers. Where the MVEM control input u is the injected fuel mass mfi and the disturbance input Ø is the throttle angle position. The feed-forward controller that correlates the steady state value between the MVEM control input mfi and the disturbance Ø will be used in the feed-forward path. In order to achieve better transient response, feed-forward and feed-back controllers will be designed as following.

A. FF controller design

The feed-forward controller will be implemented by look-up table configuration. The data of this table were determined from the MVEM. Firstly, throttle angle position value have been given to the MVEM starting from 20 to 60 degree by step 5 degree in order to cover 9 cases, secondly, the gain k has been changed for each case to adjust the air fuel ratio equal to 14.7. Finally, the suitable corresponding injected fuel mass can be determined for each throttle angle value by using (1).

$$mfi = k \times \phi \qquad (1)$$

Omar AL-Mukhtar University, Libya

978-1-4673-1558-6/12/$31.00 ©2012 IEEE

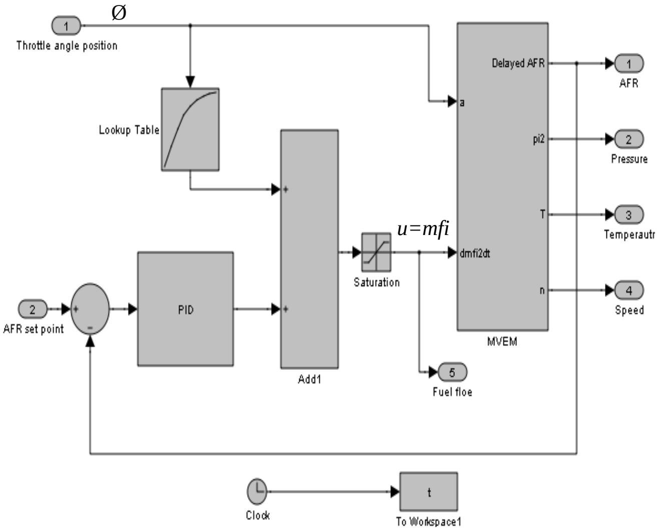


Figure 1. Simulink model for the automatic control loop for the MVEM including feed-forward and feedback controllers

B. PID controller design

In general, the transfer function of PID is illustrated in (2), where, K_p, K_i and K_d are proportional, integral and differential gains respectively.

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \tag{2}$$

Where:

$$K_i = \frac{K_p}{T_i} \tag{3}$$

$$K_d = K_p T_d \tag{4}$$

Where:

T_i: Integral time.

T_d: Derivative time.

In order to find out the PID controller parameters (K_p, Ki and K_d), many numerous tuning rules for PID controller can be found in the literature. The process of selecting the controller parameters to meet given performance specification is known as controller tuning. The rules for determining values of the K_p, T_i and T_d based on the transient response characteristics of given plant have been proposed by Zigler and Nichols [4]. All the PID parameters were determined by using the Matlab software R2009a. Equation 5 shows the transfer function of the PID controller after calculate all its parameters.

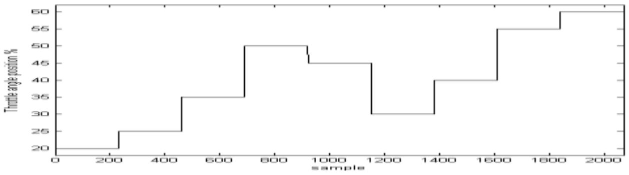
$$G_c(s) = 0.00001(1 + \frac{1}{100s} + 0.01s) \tag{5}$$

III. Evaluation of closed loop control system

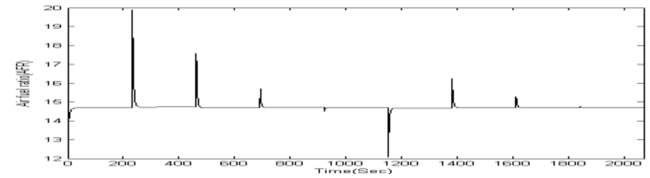
In order to evaluate the MVEM under close loop control system, a set of signal was used for the throttle angle position to obtain a representative set of input data.The range of this excitation signals was bounded between 20 and 60 degrees. This almost covers the whole throttle angle position in normal operation condition. The outputs of the PID and feed forward controllers will be used as a second input of the MVEM (see Fig. 1). Fig. 2(a ,b) illustrate the input signal of throttle angle position and the AFR output response of the MVEM close loop control obtained from the block diagram shown in Fig. 1. The AFR is to be controlled at 14.7. From the Fig. 2 it can be seen that the PID controller has good performance and the obtained results were very accurate, therefore the AFR has been controlled at 14.7.

IV. Engine modeling

The first step in the engine modeling by using RBFNN is the generation of a suitable training RAS data set of throttle angle position and setpoint of AFR. As the training data will influence the accuracy of the neural network modeling performance, the objective of experiment design on training data is to make the measured data become maximally informative, subject to constraints that may be at hand. As mentioned above, a set of random amplitude signals (RAS) were designed for the throttle angle position and AFR setpoint to obtain a representative set of input data of MVEM close loop. The sample time of 0.02 sec was used. The second step is to determine the input variables of the RBF model. The SI engine to be modeled has two input variables: throttle angle and the outputs of the PID and feed forward controller which is fuel flow rate, and four outputs: air manifold temperature, air manifold pressure, crank shaft rotary speed and air fuel ratio. The network input that generated the smallest modeling errors was selected, and has first-order for the two process inputs and third order for process output. As selected above, the RBF model has 16 inputs and 4 outputs. The hidden layer nodes have been selected as 15. Before the training, 15 centres were chosen using the K-means clustering algorithm, and the width σ was chosen using the p-nearest-neighbours algorithm. All Gaussian functions in the 15 hidden layer nodes used the same width. For training the weights W the recursive least squares algorithm [5] was applied and the following initial values were used: μ= 0.98, w (0) =$1.0\times10^{-6}\times U$ (nh×4), P(0)=$1.0\times10^{8}\times I$ (nh), where μ is the forgetting factor, I is an identity matrix and U is the matrix with all element unity, nh is the number of hidden layer nodes. Totally a data set with 7000 samples was collected from the MVEM. Before training and testing, the raw data is scaled linearly into the range of [0 1]. Fig. 3 show the model training results of the last 500 samples in the training data set and the first 500 samples in the test data set.

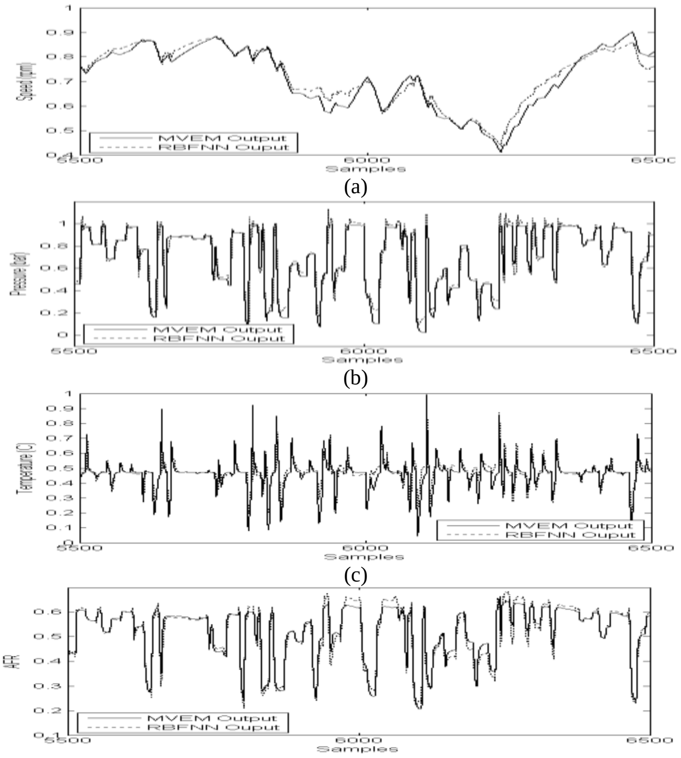


(a)



(b)

Figure 2. (a) the input signal of throttle angle position, (b) the AFR output response of the MVEM control loop



(a)

(b)

(c)

(d)

Figure 3. ,a,b,c and d. The simulation results of the speed, pressure, temperature and air fuel ratio engine model output and the RBFNN s output respectively

From Fig. 3, it can be seen that there is a good match between the two outputs with a very small error, in general. The modeling error of the training data set is smaller than the test data set. The mean absolute error (MAE) index is used to evaluate the modeling effects. For this model the MAE values of crankshaft speed, manifold pressure, manifold temperature and air fuel ratio are 0.0014, 0.0061, 0.0031 and 0.0031 respectively.

V. Simulating Faults

Before the developed method is tested on a real engine with real faults, it was tested in this research on the nonlinear simulation of SI engines, the MVEM with different faults simulated on it. One component fault, one actuator fault and four sensor faults with different levels of intensity have been investigated as practical examples of spark ignition (SI) engine faults. The component fault is air leakage in the intake manifold. The actuator fault is a malfunction of the fuel injector. The four sensor faults are malfunction of the intake manifold pressure sensor, manifold temperature sensor, crank shaft speed sensor and air fuel ratio sensor. Details of the simulation of these faults are described as follows.

A. *Component fault*

Equation (6) of the manifold pressure [6] is modified to (7) in order to collect the engine data subjected to the air leakage fault.

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

$$\dot{p}_i = \frac{T_i R}{V_i}(-\dot{m}_{ap} + \dot{m}_{at} + \dot{m}_{EGR} - \Delta l) \qquad (7)$$

where $\dot{p}_i$ is the absolute manifold pressure (bar), $\dot{m}_{at}$ is the air mass flow rate past throttle plate (kg/sec), $\dot{m}_{ap}$ is the air mass flow rate into the intake port (kg/sec), $\dot{m}_{EGR}$ is the EGR mass flow rate (kg/sec). The added term is used to simulate the leakage from the air manifold, which is subtracted to increase the air outflow from the intake manifold. $\Delta l = 0$ represents no air leak in the intake manifold. The air leakage level is simulated as 20% of total air intake in the intake manifold. This fault occurs from the sample number 3750~ 3850 in the faulty data as shown in Fig.4, and was simulated by changing the Simulink model of the MVEM.

B. *Actuator fault*

For SI engines, the target is to achieve an air–fuel mixture with a ratio of 14.7 kg air to 1 kg fuel. This means the normal value of air fuel ratio is 14.7. Because any mixture less than 14.7 to 1 is considered to be a rich mixture, any more than 14.7 to 1 is a lean mixture. Lean mixture causes the efficiency of the engine reduced, while rich mixture will cause emission increased. The fuel injector is controlled by the controller with correct amount of fuel. If the fuel injector has any fault the injected fuel amount will not be correct and affect the air/fuel ratio. Here, the malfunction of the fuel inject is simulated by reducing the injected fuel amount of 25% of the total fuel mass flow rate between the sample number 2550 and 2650 as shown in Fig.4. This fault is also simulated by changing the Simulink model of the MVEM.

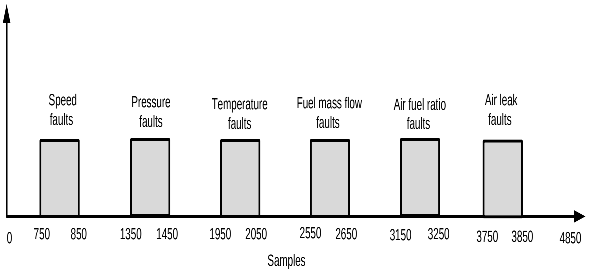


Figure 4. Distribution of the simulated faults

C. Sensor fault

The four sensor faults considered are (10, 15, 10 and 20) % changes superimposed on the outputs of crankshaft speed, manifold pressure, and temperature and air fuel ratio sensors respectively. These faults are simulated from sample numbers 750 to 850, 1350 to 1450, 1950 to 2050 and 2550 to 2650 respectively as shown in Fig.4. The faulty data for the sensors is generated using multiplying factors (MFs) of 1.1, 1.1, 1.15 and 1.2 for the above over-reading faults respectively. Faulty data are generated by the Modified MVEM with throttle angle at different values between 20^o and 60^o for all the fault conditions. The 6 states with their multiplying factors (MFs) are given in table I. The sample time is chosen as 0.02 sec.

VI. FAULT DETECTION

Fig.5 shows the information flow for the fault detection and isolation. Firstly, the 7000 samples data set of random amplitude sequence for throttle angle in the proper range and the outputs of the PID and feed forward controllers are fed into the MVEM under close loop control system. The collected four engine outputs together with the two inputs as well as their delayed values are used to train the RBF model. After training, all the six faults are simulated to the MVEM. Then, with another 4850 set of square signals of throttle angle position (see Fig. 6) and fuel flow which is output of PID and feed forward controller (see Fig. 7) fed into the MVEM under close loop control system, the model prediction error and the filtered residual are generated for fault detection. With the six faults simulated from samples 750 to 3850 as shown in Fig.4.After a low-pass filter is used the filtered prediction errors are shown in Fig.8 (b, c, d & e). The first 500 samples of data set which mean the beginning 10 second of engine operation has been ignored because contain noise signals. The first filtered model prediction error of air fuel ratio is shown in Fig.8, b. The second, third and fourth for air manifold temperature, pressure and engine speed are showed in Figs.8(c, d, e) respectively. In these Figs. the samples 0 to 750 are data without faults. Including them is to show the prediction error is under the selected threshold in “no fault case”. Now it is evident that all simulated faults have a significant reflection on the model prediction errors. A threshold is chosen for each prediction error and is also displayed in these Figs. Moreover, Fig. 8, f shows the residual error (*re*) which is generated by (8).

$$re = \sqrt[2]{e_n^{2} + e_p^{2} + e_t^{2}} \qquad (8)$$

Where e_n, e_p and e_t are the error vectors of the speed, pressure and temperature respectively between the engine model and the RBF neural network.

TABLE I. THE 6 FAULTS STATES AND MULTIPLYING FACTORS

No	Fault Name	multiplying factors (MFs)
1	Air Leak 20%	
2	Injected fuel mass flow 25%	
3	Speed sensor 10% over reading	1.1
4	Pressure sensor 15% over reading	1.15
5	Temp. sensor 10% over reading	1.1
6	Air fuel ratio 20% over reading	1.25

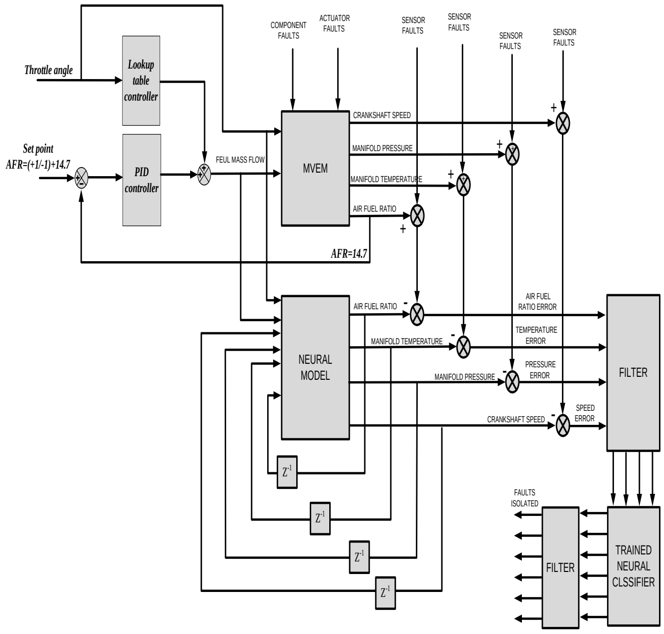


Figure 5. The information flow for the fault detection and isolation

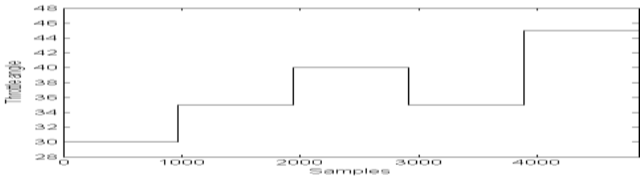


Figure 6. Square signals of throttle angle position

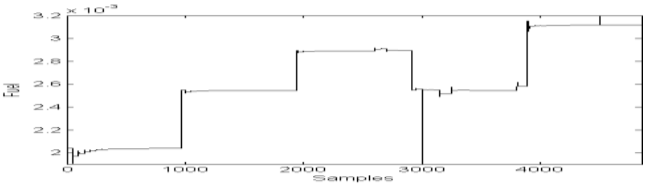
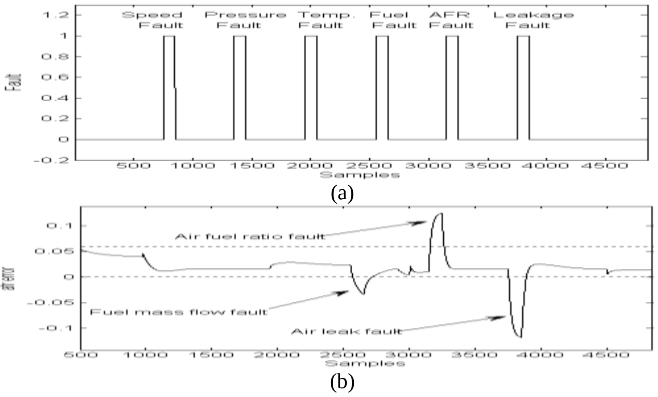
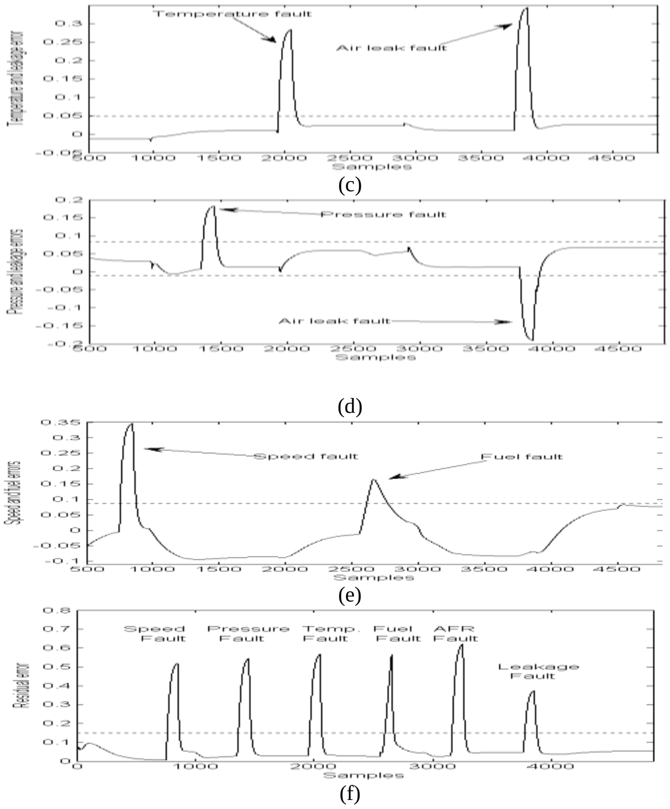


Figure 7. fuel flow rate (output of PID and feed forward controllers)



(a)

(b)



(c)

(d)

(e)

(f)

Figure 8. (a)simulated faults, (b), (c) , (d) and (e) Filtered model prediction error of air fuel ratio, air manifold temperature, air manifold pressure and engine speed , (f) Filtered residual

VII. Fault isolation

As have been seen in Fig.8, f, though all simulated faults cause a significant deviation in the residual, this can be used to detect fault but cannot be used to isolate fault. When a fault occurs, only its associated output of the fault diagnosis system has a reflection, while all the other outputs should be insensitive to this fault, then the fault can be isolated from the other possible faults. In this research there are 6 possible faults but only four modeling errors. It can be concluded from Fig.8 that if only the first model output (air fuel ratio) goes over the threshold while the other three outputs remain under the thresholds, then it must be air fuel ratio sensor has fault. If second model output (temperature) goes over the threshold while the other three outputs remain under the thresholds, then it must be temperature sensor has fault. Similarly, if only the third model output (pressure) goes over the threshold while the other three outputs remain under the thresholds, then it must be the pressure sensor has fault. Moreover, if only the fourth model output (speed) goes over the threshold while the other three outputs remain under the thresholds, then it must be the speed sensor has fault. If both the first model output (air fuel ratio) and the forth model output (speed) go over the thresholds while the other two outputs remain under the thresholds, then it must be the fuel injector has fault. Finally, If the first, second and third model outputs go over the thresholds while the fourth output remain under the threshold, then it must be the air leak occurs. Therefore, to achieve a clear isolation among all possible faults, another RBF network is employed as a fault classifier. The classifier has four inputs each receiving one of the four modeling errors, and has seven outputs with one representing "no fault case" and the other six representing the six different faults. The classifier is trained in the following way. Collect 7 sets of data with first set without fault and the other six sets each with one fault only. For each data set of the seven, the target of the training for the output corresponding to the contained fault is set to "1", while the targets for the other outputs set to "0". Totally 4850 RAS data samples were collected with first 750 without fault and each set of 600 samples with one of the five faults, the last sixth fault occurred during the last set of 1100 samples. These data are fed into the new RBF model and the generated four modeling errors are fed into the RBF classifier to train it, with the targets given as described above. After training, the classifier is tested with the similar arrangement of data, totally 4850 data samples which are the modeling errors obtained from the RBF of fault detection and MVEM under close loop control system (see Fig. 8, b, c, d and e). The first 750 samples are fault-free, followed by six data sets. The first five data sets have 600 samples and having a single fault, the sixth data set has 1100 samples and has the last fault. Between any two of these six faulty data sets insert 500 fault-free samples and the final 1000 samples are fault-free, so that the residual rising time and disappearing time can be observed. The data samples with associated fault types are listed in Table II. Similar to the first RBF network, the centres and widths are also selected using the K-means clustering algorithm and the *P*-nearest centres method respectively. The network weights are trained using the recursive Least Squares algorithm with its parameters set as μ= 1.0, w (0) =1.0×10^{-6}×U (nh×6), P (0) =1.0×105×I (*nh*). The number of the hidden layer nodes was tried several numbers and the one giving minimum training error was chosen and was 250. Figs. 9~14 show the test result after filtering. The isolation thresholds are chosen as shown in the Figs.

TABLE II. Data samples and fault types

Data samples	Fault types
1 ~ 750	No fault
751 ~ 850	Speed sensor fault
801 ~ 1350	No fault
1351 ~ 1450	Pressure sensor fault
1451 ~ 1950	No fault
1951 ~ 2050	Temperature sensor fault
2051 ~ 2550	No fault
2551 ~ 2650	Fuel injector fault
2651 ~ 3150	No fault
3151 ~ 3250	Air fuel ratio fault
3251 ~ 3750	No fault
3751~3850	Air leak fault
3851~4850	No fault

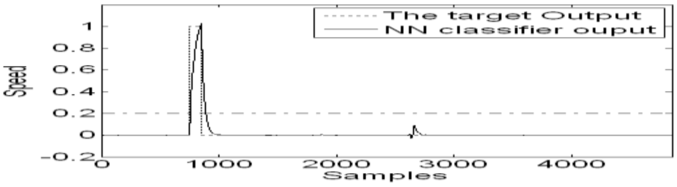


Figure 9. Filtered first output of fault classifier

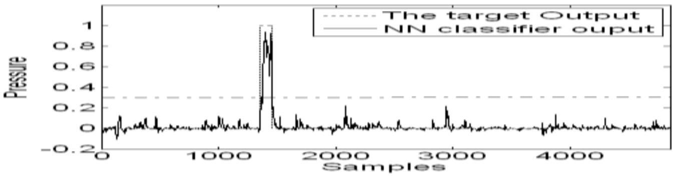


Figure 10. Filtered second output of fault classifier

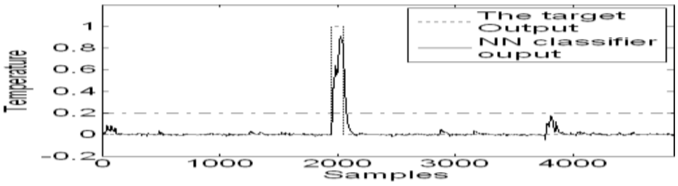


Figure 11. Filtered third output of fault classifier

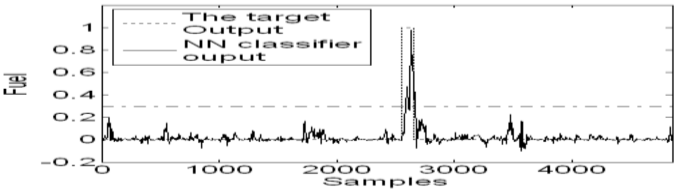


Figure 12. Filtered forth output of fault classifier

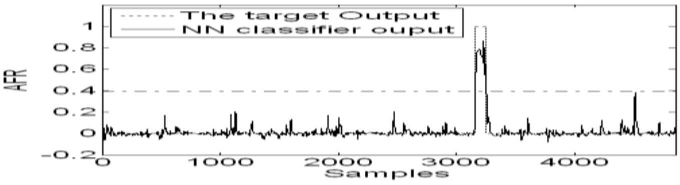


Figure 13. Filtered fifth output of fault classifier

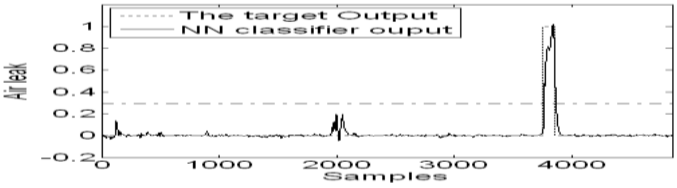


Figure 14. Filtered sixth output of fault classifier

VIII. Discussion of Simulation Results

A. Training the Neural Network model.

The simulation results of training and testing by using 15 hidden nodes were very good and in general, a good prediction between the engine model output and the RBF neural network output was achieved. From Fig. 3, it can be seen that the mean absolute error between the engine speed output and the RBFNN is very small.

B. Detection of the Sensor, Component and Actuator Faults.

The Fig. 8 (b, c, d and e) show the test results of the fault detection of air fuel ratio/fuel flow/leakage, temperature/air leak, pressure/air leak, and speed/fuel injection respectively after filtering with 15 hidden nodes. It can be seen from Figs. after filtering operation that all kind of faults were detected clearly. The error values were between 0.05 and -0.1 except the samples in which faults occur. The detection thresholds were chosen as 0.09 for crankshaft speed / fuel injection, (-0.01/+0.08) for manifold pressure/air leak, 0.05 for manifold temperature/air leak and (0/0.06) for air fuel ratio.

C. Isolation of the Sensor, Component and Actuator Faults.

In this section, another RBFNN called neural classifier were used in order to isolate all kinds of the faults. This neural was received error signals between the MVEM and the RBFNN model outputs. 50, 150 and 250 hidden nodes are used in order to try to obtain good simulation results. From the Figs. of the results, we found 250 hidden nodes after filtering operation were the best case and the simulated faults can be clearly isolated (see Figs. 9~14). The isolation thresholds are chosen as shown in the Figs. in case 250 hidden nodes.

IX. Conclusions

In this research, the MVEM under close loop control including feed-forward and feed-back controllers was used. Look up table has been used as a feed-forward controller and PID used as feed-back controller. The data of the Look up table were determined by using the MVEM and the data of PID were calculated by Zigler and Nichols methods by using the Matlab software R2009a. The AFR output response of the MVEM close loop control was controlled at 14.7. Four sensor faults (intake manifold pressure, temperature, crankshaft speed and air fuel ratio), one component fault (leakage in the intake manifold) and one actuator fault (injected fuel mass flow) have been simulated. Two RBFNNs were used, the first one to model engine dynamics and the second one to isolate the sensor faults, component fault and actuator fault from the modeling errors. By using *p* – Nearest Neighbours method and K-means algorithm the width in hidden layer nodes of the RBF neural network σ and the centres c are calculated for both RBFNNs. The recursive least square algorithm was applied for training the weights *w* of the RBF neural networks. The proposed method can detect and isolate the faults and from the Figs. of simulation results, it can be seen that the methods were able to detect and isolate the faults.

References

[1] R. Isermann, “Model-based fault-detection and diagnosis status and applications,” Annual Reviews in Control, vol.29, pp.71–85 ,2005.

[2] D. Capriglione, C. Liguori, C. Pianese, and A. Pietrosanto, “Analytical redundancy for sensor fault isolation and accommodation in public transportation vehicles,” IEEE Trans. on Instrumentation and Measurement, vol. 53, No.4, pp. 993-999, 2004.

[3] A. Hamad, D. Yu, J.B Gomm,. and S. Mahavir, “Automotive engine fault detection and isolation via radial basis function neural network,” Proceeding of 16th international conference on automation & computing, editors Xun Chen, Jihong Wang ,Birmingham, UK.,pp.232-237 , ISBN978-0-9555293-6-8, 2010.

[4] K. Ogata, Modern control engineering. Prentice-Hall international, inc, 1997.

[5] Y. J. Zhai and D. Yu, “Radial Basis Function Based Feeddback Control for Air Fuel Ratio of Spark Ignition,” Proc. IMACE Joural of Automobile Engineering, Vol. 222, pp. 415-428, 2007.

[6] E. Hendricks, D. Engler, and M. Fam, “ A Generic Mean Value Engine Model for Spark Ignition Engines,” Proceedings of 41st Simulation Conference, Denmark, DTU Lyngby, 2000.