# Failure detection of closed-loop systems and application to SI engines

L.-F. Deng*, Y.-W. Shi*, L.-X. Zhu* and Dingli Yu[$]

* College of Communication Engineering, Jilin University, Changchun, Jilin Province

$ Control Systems Research Group, School of Engineering, Liverpool John Moores University

Byrom Street, Liverpool L3 3AF, U.K.

---

**Abstract.** In recent years, Fault detection and isolation (FDI) have become one of the most important aspects of automobile design. FDI for internal combustion engines based on the open-loop system model was investigated. However, due to some differences between open-loop and closed-loop systems, the FDI scheme developed under open-loop could not be directly applied, without further adjustment, to practical engines in vehicles. In this paper, a new FDI scheme is developed for automotive engines under closed-loop control. The method uses an independent radial basis function (RBF) neural network model to model engine dynamics, and the modelling 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 by classifying fault characteristics embedded in the modelling errors. The performance of the developed scheme is assessed using an engine benchmark, the Mean Value Engine Model (MVEM) with Matlab/Simulink. Six faults are simulated on the MVEM, including four sensor faults on manifold pressure, manifold temperature, crankshaft speed and air fuel ratio, one component fault of air leak, and one actuator fault of malfunction of fuel injector. 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.

## 1. INTRODUCTION

Fault detection and isolation (FDI) for automotive engines have been investigated for over two decades. A fault is definded as any type of malfunction of components, actuators and sensors, which may happen in a system and will degrade the system performance or stability but not cause catastropy. Fault detection is the procedure that reports the occurrence of a fault and occuring time, normally by collecting input/output data of the system, and then checking them against a certain set of ralationships among these data [ ]. Fault isolation is to determine, after a fault is detected, which fault occurs among all the possible faults pre-defined. Further investigation about fault size and location is referred to as fault diagnosis. Compensation for the fault effects by re-configurating the control system so that the degraded performance or stability could be recovered or maintained till the system is convinient for repair, is called fault tolerant control. Engine FDI is important because it is an ecential method to enhance reliability of engines, and consequently vehicles. Also, the polution caused by fault (for example, the degraded air/fuel ratio caused by mvlfunction of sensors) can be greatly reduced.

FDI for automotive engines has been investigated for more than two decades. Gertler (1993) and his co-workers attemted using linear Parity space method to detect simulated faults. Simulated faults and experiments have been done. But the results were not very satisfactory due to the severe nonlinearities in engine dynamics. Yu, et al (1991) have introduced architecture for a special purpose diagnostic processor, designed specifically to interact as a co-processor with the on-board controller. The processor considered the structure of model-based failure detection algorithms and the need for simulating the dynamics of vehicle subsystems in order to monitor their performance, and processes the information provided by on-board sensors to diagnose sensor or actuator malfunctions. Blanke, et al. (1995) introduced an electro-mechanical position servo, used in the speed control of large diesel engines, as a benchmark for model based fault detection and identification. Isermann (2005) proposed model-based fault-detection and diagnosis methods for some technical processes. The goal was to generate several symptoms indicating the difference between nominal and faulty status.

On-line sensor fault detection, isolation, and accommodation in automotive engines have been studied by Capriglione, et al. (2003, 2004 and 2007). Their paper described the hybrid solution, based on artificial neural networks (ANN), and the production rule adopted in the realization of an instrument fault detection, isolation,

and accommodation scheme for automotive applications. However, their methods used ANN just as classifiers rather than dynamic models. Therefore, the application of the method would not be straight forward and need adjustment for each individual engine. Also, the accuracy of the fault detection reported in their papers is very rough, and further improvement is needed for possible real applications. Fault detection for modern diesel engines using signal- and process model-based methods have been proposed by Kimmich, et al. (2005). Their contribution showed a systematic development of fault detection and diagnosis methods for two system components of diesel engines, the intake system and the injection system together with the combustion process. The residuals were generated by applied semiphysical dynamic process models, identification with special neural networks, signal models and parity equations. The deflection of the residuals allowed the detection and diagnosis of different faults.

It is noticed in the literature that most reported research on FDI of internal combustion engines are based on engine itself, in other words, most methods were developed and evaluated based on the engine that is not under closed-loop control as in the practical applications. Due to a number of differences between open-loop engines and closed-loop controlled engines, the FDI methods developed on the open-loop engines will find difficulties and problems when they are tested with the engines under closed-loop control. In further, these problems also appear when these methods were tried to be applied to real engine systems. The main differences between open-loop engine and engine under closed-loop control are lies on the following points.

Firstly, the neural network model of the engine air path needs to be trained with engine input/output data. In order to excite the engine dynamics on the whole range of frequency, for open-loop engine a random amplitude sequence (RAS) is used. However, as the input signal for engine under closed-loop control is the controller output and cannot be designed, the persistent exciting will not be achieved. One method is to superimpose a pseudorandom binary sequence (PRBS) onto the control signal. To excite nonlinear dynamics of a nonlinear dynamic system such as engines, both amplitude and frequency of the excitation signal should be considered for their wide range. As the diverged amplitude can be achieved by the variable amplitude of the control signal, while the diverged frequency achieved by the PRBS, the above stated signal will be used in this research.

Secondly, the RAS signal applied to set-point of the closed-loop system has been considered in this

research, and then the generated control signal is applied to the engine for data excitation. This is from the fact that in practice the disturbance, the throttle angle change, is often a step change, which can be simulated by a RAS signal. On other hand, the control signal would be a filtered signal (as the controller could be seen as a filter) and lost the deep edge of the step signal, so the high frequency part would be lost. However, in practice it is just this type of signal dominated the operation process, and therefore, the training of the NN model with these data would be reasonable.

Thirdly, the output of some sensors may be used as feedback signal in the closed-loop control. This will change the signal pattern completely. When a sensor fault occurs, the sensor output is feedback to change the error signal, so that the engine output is regulated to a value such that the sensor output nearly equivalent to the set-point. This makes the deviation of the sensor output from nominal value quickly vanishes, and consequently the detection on this signal would become more difficult.

Fault detection and isolation for MVEM open loop control system (see Figure 2) were achieved in previous paper of the authors (Adnan Hamad et al, 2010). Good simulation results were obtained and all the simulated faults were clearly detected and isolated in dynamic conditions throughout the engine operation range. However, in the practice the engine does not work as an open loop control system. Automotive engine considered as a close loop control system with feed-back control.

In order to solve all these problems which were happened when the open loop control system were used for fault detection and isolation and to obtain good simulation results, close loop control system will be used for MVEN. It's expected very good fault detection and isolation simulation results will be obtained from the developed close loop control system. Excitation input signals (throttle angle and fuel mass flow) were used in case open loop MVEM. These input signals were random amplitude signal (RAS) and covered all engine operation range. The output obtained from this kind of control was changed rapidly as same as the RAS input signals. However, if RAS used in the close loop MVEM then the output will follow the reference input signal (set-point) in order to try to keep control for AFR at 14.7. Then the logic result for this situation, the AFR output will be changed rapidly around 14.7. In order to solve this problem, another kind of data will be used to cover the engine operating range. This input data which is throttle angle will start from 20 to 60 by step 5, each step has 20 samples. The second input will be the output of the feed-back controller.

In this paper, a new fault detection and isolation method will be implemented by using mean value engine model when this model is under close loop control system. An independent RBFNN model is used to model a dynamic system. Feed-back (FB) and feed-forward (FF) control methods will be applied to the mean value engine model. Firstly, fuel injection value corresponding to angle position value will be defined by using MVEM. Secondly, PID controller will be used in the feed-back to adjust the difference between the requested and the actual air fuel ratio by compensating the feed-forward controller output. Thirdly RBF neural network models will be used to predict MVEM outputs. The model prediction error is used as residual to report occurrence of fault. Then, another neural network is trained as a classifier to isolate different faults using residual vector according to the different features on the residual vector by different faults. The K-means clustering algorithm is used to choose the centres of RBF network. Recursive least squares (RLS) algorithm is used to update for each new sample the parameter matrix *W*. The proposed method is applied to air path dynamics of the IC engines using a well-know benchmark simulation model: mean value engine model (MVEM). Different types of fault were simulated on the MVEM, and then the input/output data were used to detect and isolate these faults. This type of neural networks have been chosen because it has a simple structure, easy to train and it is capable of approximating a nonlinear function to any desired degree of accuracy, it is used as an independent model of a dynamic system. A set of very good results has been achieved, indicating that the developed method has a high potential to be used in practice of new generation vehicles. Model and model artificial neural networks provide an excellent mathematical tool for dealing with non-linear problems. They are especially useful in situations when there is no mathematical model of the process considered, so the classical approaches such as observers or parameter estimation methods cannot be applied.

## 2. Engine Dynamics and the Mean Value Engine Model

The configuration of an internal combustion (IC) spark ignition (SI) engine is shown in Fig. 3 (Beham, 2004). Simply, the air is flowed through the air manifold under the control of the throttle valve into the cylinder, where it is mixed with fuel injected into the cylinder by fuel injector. Then, the chemical energy of the fuel is converted kinetic energy by piston and crank shaft to drive the car. The corresponding engine variables in Fig.3 are listed in Table 1.

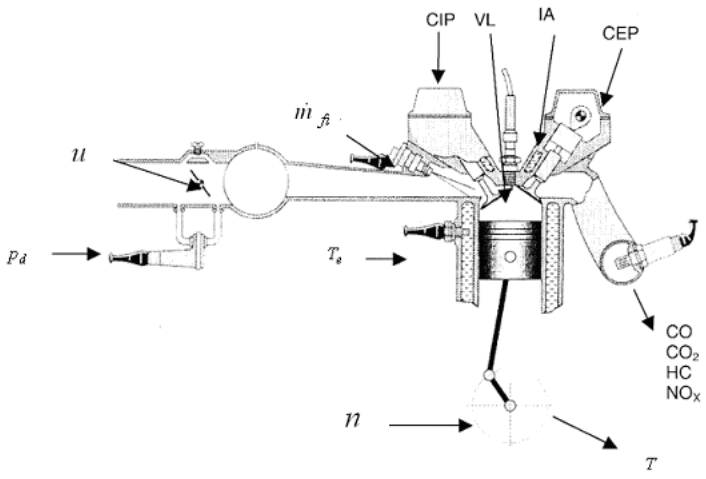



Fig.3 Schematic of SI engine and its air path

Table 1 Variables in Fig.1

| **Input Variables** | | **Output Variables** | |
|---|---|---|---|
| Camshaft intake position | CIP | Torque | *T* |
| Camshaft exhaust position | CEP | Carbon monoxide | CO |
| Valve lift | VL | Carbon dioxide | CO2 |
| Ignition angle | IA | Hydrocarbon | HC |
| Throttle angle | *u* | Nitrogen oxide | NOx |
| Fuel injection | $m_{fi}$ | Engine speed | *n* |
| Differential pressure | $p_d$ | Engine temperature | *Te* |

As the above variables take different values in the four strokes during engine operation and change very fast, the engine model must be very complex to reflect these quick changes. However, the four strokes are

repeated every cycle, so that the average value of these variables over a single cycle can be used in the model. A mean-value engine model (MVEM) developed by Elbert Hendricks (2000) in Fig.4, which is a widely used benchmark model for engine modelling and control, is adopted in this research as the nonlinear simulation of the SI engines. The dynamics of the MVEM include three sub-systems: air manifold filling dynamics, crankshaft speed dynamics and fuel injection dynamics (see Figure 4).

### 2.1. Major Engine Control Variables

Due to the increasing requirements of governments and customers, the main objective of SI engine development is to generate a power output as high as possible, while at the same time keeping fuel consumption and exhaust emissions down to a minimum in order to comply with the requirements of emissions-control legislation. To satisfy these requirements, many variables such as engine speed, engine torque, ignition angle, injection timing, emission gas and so on need to be controlled. It is very difficult to achieve satisfactory control performance for these variables because they are severely non-linear and complicatedly related to each other. Moreover, car engines have several different operating modes including start up, idle, running and braking. Therefore, engine dynamics are highly non-linear and multivariable, which makes engine dynamic fault detection more complex and difficult (S. W. Wang et al, 2006 and Balluchi, et al,2000).

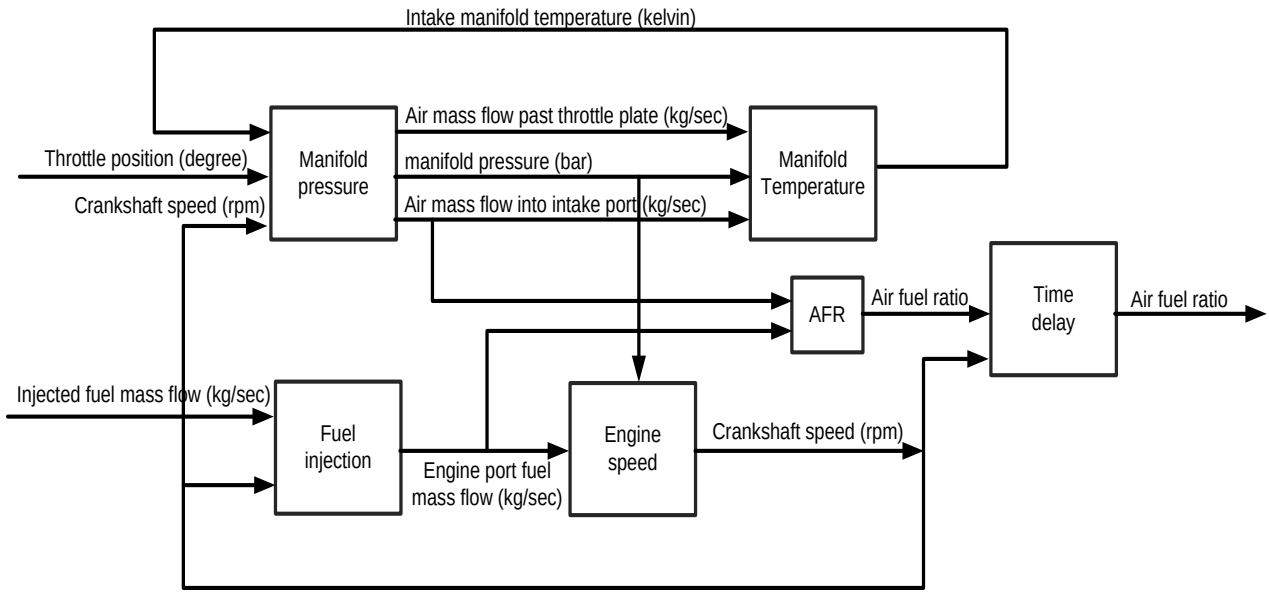


Fig.4 Mean value engine model

### 2.2. Manifold Filling Dynamics

It includes two nonlinear differential equations: one for the manifold pressure and the other for the manifold temperature. The manifold pressure is described as

$$\dot{p}_i = \frac{RT_i}{V_i}\left(-\dot{m}_{\mathrm{at}} + \dot{m}_{ap} + \dot{m}_{\mathrm{EGR}}\right) \tag{1}$$

Where $\dot{\mathrm{p}}_{\mathrm{i}}$ is absolute manifold pressure (bar), $\dot{\mathrm{m}}_{\mathrm{at}}$ is air mass flow past throttle plate (kg/sec), $\dot{\mathrm{m}}_{\mathrm{ap}}$ is air mass flow into intake port (kg/sec), $\dot{\mathrm{m}}_{\mathrm{EGR}}$ is EGR mass flow (kg/sec),$T_i$ is intake manifold temperature in Kelvin, $V_i$ is (manifold & port passage) volume ($m^3$) and $R$ is gas constant ($287\times10^{-5}$). The manifold temperature dynamics are described by the differential equation (Hendricks, et al, 2000)

$$\dot{T}_i = \frac{R\ T_i}{P_i V_i}\left[-\dot{m}_{ap}(k-1)\ T_i + \dot{m}_{at}\ (k\ T_a - T_i) + \dot{m}_{EGR}\ (k\ T_{EGR} - T_i)\right] \tag{2}$$

### 2.3. Crankshaft Speed Dynamic

The crankshaft dynamics is derived using conservation of rotational energy on the crankshaft (Hendricks, et al, 2000).

$$\dot{n} = -\frac{1}{In}\Big(P_f(p_i, n) + P_p(p_i, n) + P_b(n)\Big) + \frac{1}{In} H_u \eta_i(p_i, n, \lambda)\dot{m}_f(t - \Delta\tau_d) \tag{3}$$

Both the friction power $P_f$ and the pumping power $P_p$ are related with the manifold pressure *pi* and the crankshaft speed *n*. The load power *Pb* is a function of the crankshaft speed *n* only. The volumetric efficiency $\eta_i$ is a function of the manifold pressure *pi*, the crankshaft speed *n* and the air/fuel ratio $AFR$. Where *I* is the scaled moment of inertia of the engine and its load and where the mean injection/torque time delay has been taken into account with the variable $\Delta\tau_d$.

### 2.4. Fuel injection dynamics

According to the identification experiments with an SI engine carried out by Hendricks et al. (Hendricks, et al, 2000), the fuel flow dynamics could be described as

$$\ddot{m}_{ff} = \frac{1}{\tau_f}(-\dot{m}_{ff} + X_f \dot{m}_{fi}) \quad (4)$$

$$\dot{m}_{fv} = (1 - X_f)\, \dot{m}_{fi} \quad (5)$$

$$\dot{m}_f = \dot{m}_{fv} + \dot{m}_{ff} \quad (6)$$

Where $\ddot{m}_{ff}$ is fuel film mass flow, $\dot{m}_{fi}$ is injected fuel mass flow, $\dot{m}_{fv}$ is fuel vapor mass flow. The model is based on keeping track of the fuel mass flow. The parameters in the model are the time constant $\tau_f$ for fuel evaporation, and the proportion $X_f$ of the fuel which is deposited on the intake manifold or close to the intake valves.

## 3. Control structure for MVEM

The Figure 5 shows the block diagram for 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 controller will be designed as following.

### 3.1 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 by used Figure 6. 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 equation 7. Table 2 shows the throttle angle position with the corresponding injected fuel value and the gains for each case.

$$mfi = k \times Ø \quad (7)$$

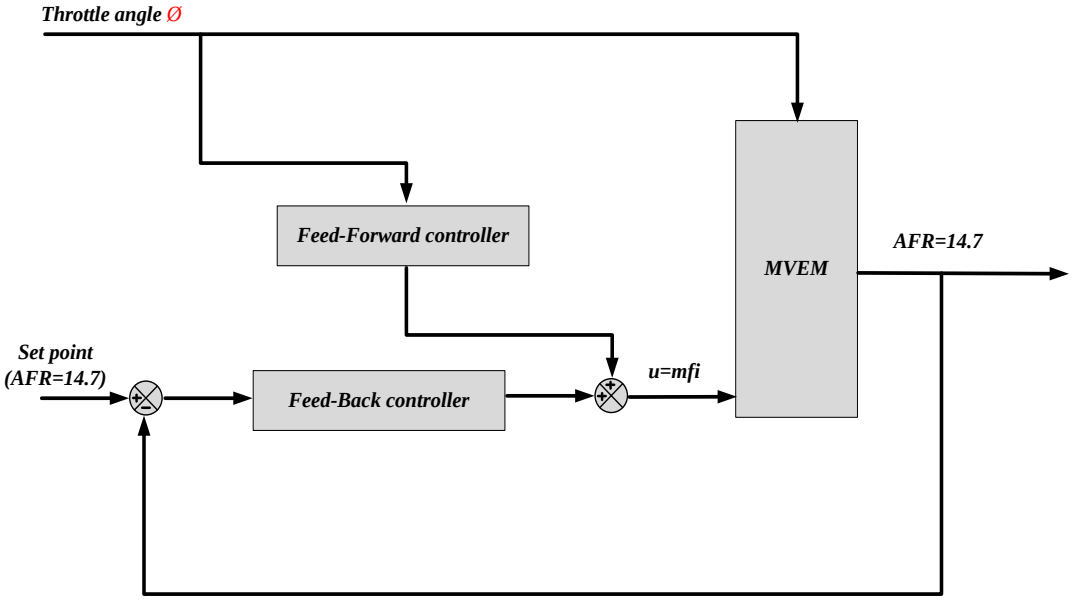



Figure 5: block diagram for the automatic control loop for the MVEM including feed-forward and feedback controllers

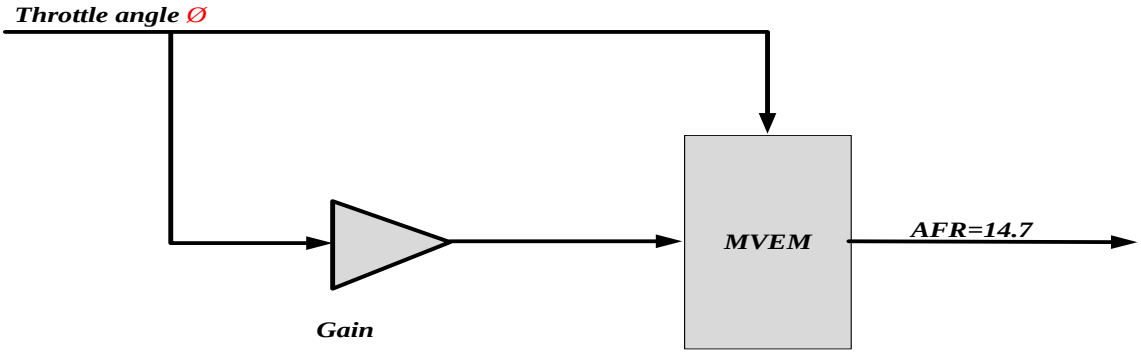



Figure 6: MVEM to define the corresponding injected fuel (mfi)

Table 2: Throttle angle position and its corresponding of injected fuel

| **Throttle angle position *(Ø)*** | **Gain (k)** | **Corresponding injected fuel (mfi)** | **Requested AFR** |
|---|---|---|---|
| 20 | 0.00003144 | 0.0006288 | 14.7 |
| 25 | 0.00005452 | 0.0013630 | 14.7 |
| 30 | 0.00006823 | 0.0020469 | 14.7 |
| 35 | 0.00007282 | 0.0025487 | 14.7 |

| | | | |
|---|---|---|---|
| 40 | 0.00007237 | 0.0028948 | 14.7 |
| 45 | 0.00006933 | 0.0031198 | 14.7 |
| 50 | 0.00006508 | 0.0032540 | 14.7 |
| 55 | 0.00006083 | 0.0033456 | 14.7 |
| 60 | 0.00005682 | 0.0034092 | 14.7 |

### 3.2 PID controller design

In general, the transfer function of PID is illustrated in equation 8, where, $K_p$, $K_i$ and $K_d$ are proportional, integral and differential gains respectively.

$$G_C(s) = K_p + \frac{K_i}{s} + K_d s \quad (8)$$

Where:

$$K_i = \frac{k_p}{T_i} \quad (9)$$

$$K_d = K_p T_d \quad (10)$$

Where:

$T_{i:}$ Integral time.

$T_{d:}$ Derivative time.

In order to find out the PID controller parameters ($K_p$, $K_i$ 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 (Ogata, 1997). All the PID parameters were determined by using the Matlab software R2009a. Equation 11 shows the transfer function of the PID controller after calculate all its parameters.

$$G_C(s) = 0.00001\left(1 + \frac{1}{100s} + 0.01s\right) \quad (11)$$

### 4. Engine modelling using RBF neural network

The radial basis function (RBF) neural network is chosen in this study to model the SI engines, because the

weights of RBF are linearly related to the objective function, so that any linear optimisation algorithm can be used to train the network weights and the training is very fast. The structure of RBF network is shown in Fig.7.

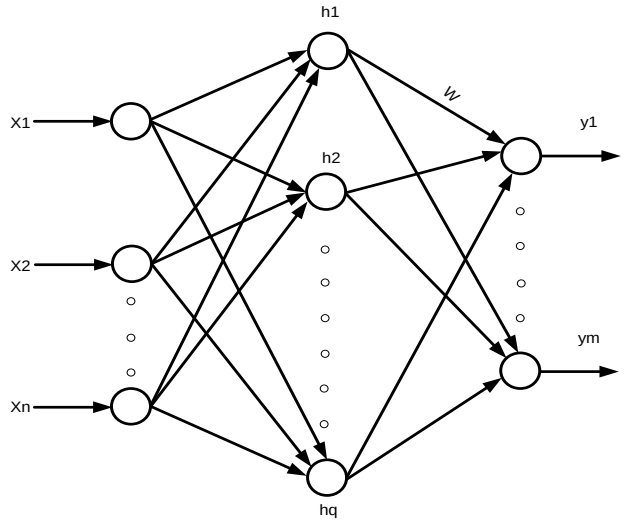



Fig. 7 The RBF network structure

When the Gaussian function is chosen, the output of the network can be calculated according to (12)-(13).

$$y = W * \phi + b \tag{12}$$

$$\varphi_i = e^{-\frac{\|x-c_i\|^2}{\sigma^2}}, \qquad i = 1, \cdots, p \tag{13}$$

where $W$ is the weight matrix, $b$ is the bias vector, $x$ is the input vector, $c_i$ is the $i^{th}$ centre vector of the same dimension of x, and sigma is the width of the Gaussian function.

**4.1 Independent RBF neural network model**

There are two different modes of modelling a dynamic system using neural networks, one is the dependent mode and the other is the independent mode as defined in [ ]. The structures of the two modes are displayed in Fig.8.

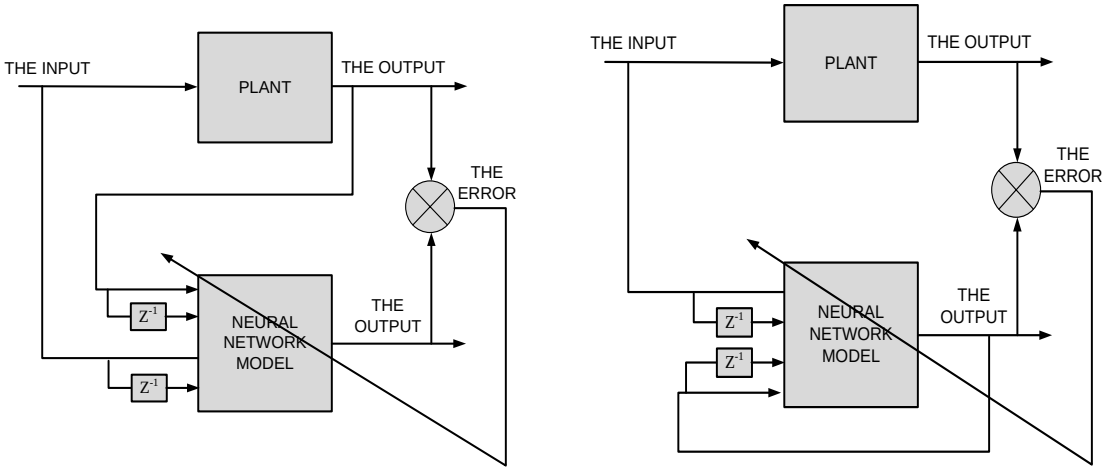



Fig. 8 The structure of the two modes: left ---- dependent mode, right ---- independent mode

It is seen that in the dependant mode the process output is used as part of network inputs. Then, the model is dependent on the process and cannot run alone. On the contact, the model output is used, instead of process output, as part of model input in the independent mode. Then, such a model can run independent of the process. The model of the dependent mode can predict the process output for one-step-ahead only, while the independent model can predict for infinity steps as long as the input is available. The advantage of the dependent model is that it is easy to be trained for accurate one-step-ahead prediction, but one-step-ahead prediction is very limited for applications. On the other hand, the independent model, though it is difficult to train, can be used as simulation model or used in Model prediction control for multi-step-ahead prediction. The above features have been experienced by the authors in the past research [ ].

When the network is used as a healthy process model to predict healthy process output, the dependent model output will also be affected if a fault occurs in the process and affects the process output. This is because the affected process output is used as the input of the model, so that the model output will also be affected. Consequently, the error between the process and model output as the residual will not be sensitive to the occurrence of the fault. This is demonstrated in Fig.9.

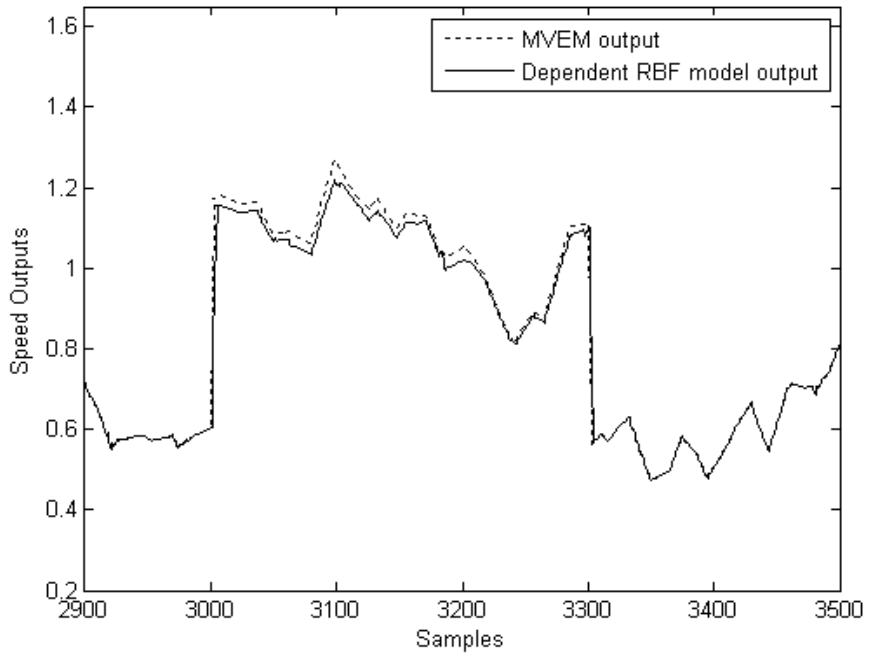



Fig.9 Comparison of engine speed from MVEM output and a dependent RBFNN model prediction

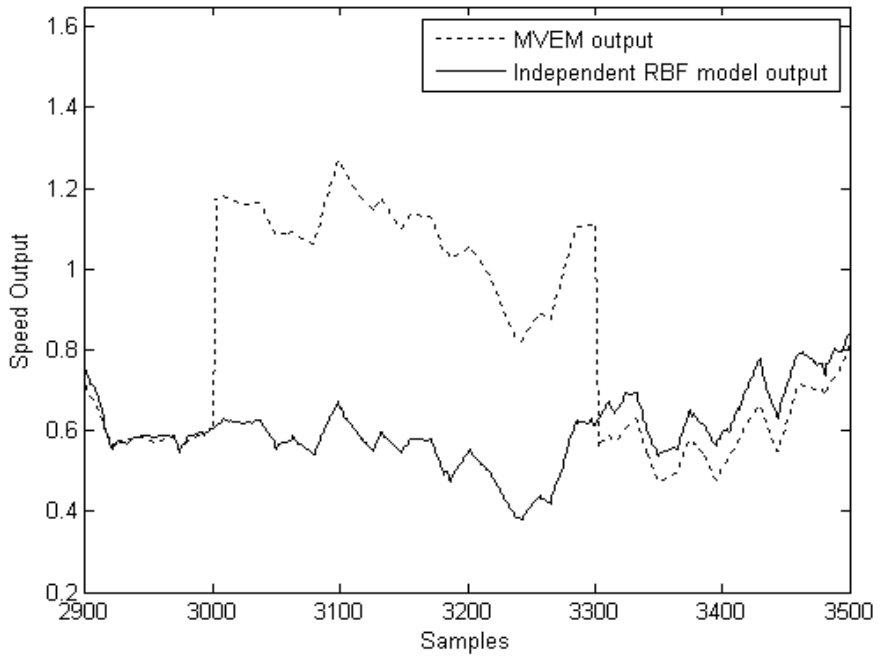



Fig. 10 An independent RBFNN model and MVEM outputs for engine speed

The same output with the prediction by an independent model is displayed in Fig.10. Obviously, the modelling error of the independent model is very sensitive to the fault, though the prediction after the fault is not as accurate as before.

### 4.2 Evaluation of closed 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 output of the PID controller will be used as a second input of the MVEM (see figure 5). Figure 11 illustrate the input signal of throttle angle position. Figure 12 shows the AFR output response of the MVEM close loop control obtained from the block diagram shown in Figure 5. The AFR is to be controlled at 14.7. From the Figure 12 it can be seen that the obtained results were very accurate and the AFR equal to 14.7. The other outputs of MVEM close loop control which are crankshaft speed, manifold pressure and temperature. All these outputs will be used as an initial value for the RBFNN in order to model the MVEM by using neural network.

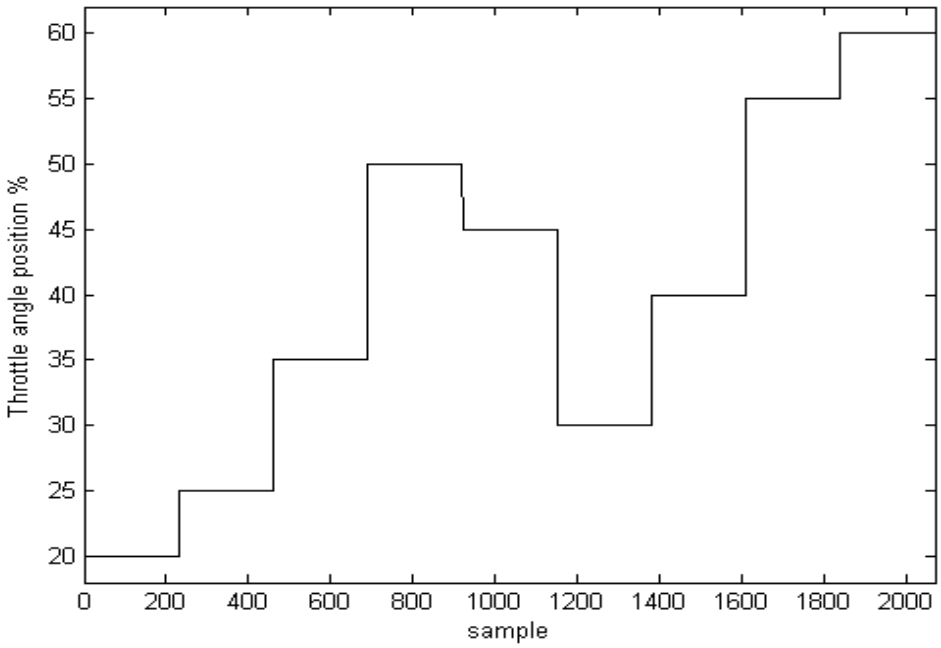



Figure 11 the input signal of throttle angle position

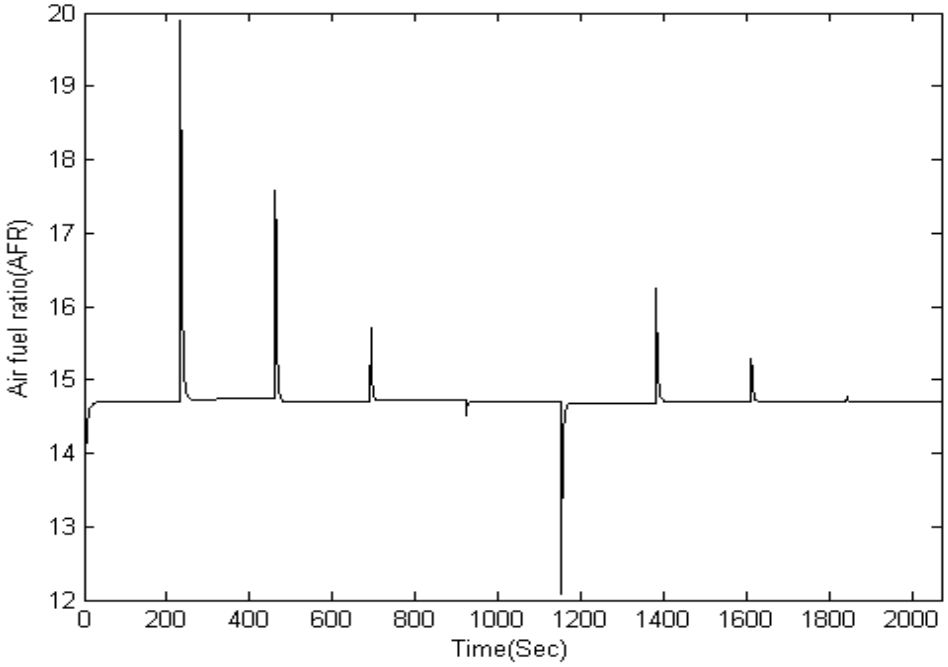



Figure 12 the AFR output response of the MVEM control loop

From the Figure 12 it can be seen clearly that the PID controller has good performance. The biggest overshot can be seen in the first step, it is about 38% overshot, and then the PID controller reduced this overshot to 14.7 quickly. In The steps 4 and 5, it can be seen two drops. Finally the PID controller has kept the setpoint in 14.7.

**4.3 Engine modelling**

The first step in the engine modelling by using RBFNN is the generation of a suitable training data set. As the training data will influence the accuracy of the neural network modelling 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 to obtain a representative set of input data. The sample time of 0.1 sec was used.

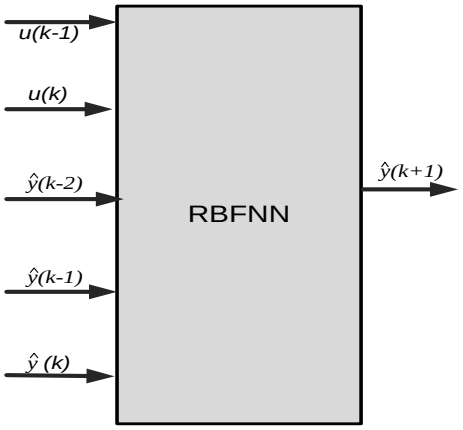



Fig. 13 RBFNN model structure

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 output of the PID controller which is fuel flow rate, and four outputs: air manifold temperature, air manifold pressure, crank shaft rotary speed and air fuel ratio. According to the dynamics in (1)-(6) and modeling trials, the network input that generated the smallest modeling errors was selected as first order for process input and second order for process output as shown in Fig.13.

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 (Zhai, *et al*., 2007) was applied and the following initial values were used: $\mu$= 0.98, $w(0) = 1.0\times10^{-6}\times U_{(nh\times4),}$ $P(0)=1.0\times10^{8}\times I_{(nh)}$, where $\mu$ is the forgetting factor, $I$ is an identity matrix and $U$ is the matrix with all element unity, $n_h$ is the number of hidden layer nodes.

Totally a data set with 2070 samples was collected from the MVEM. Before training and testing, the raw data is scaled linearly into the range of [0 1] using the following formulae.

$$x_{scaled} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{14}$$

Figs.14~17 show the model training and validation results of the 500 samples in the training data set. It can

be seen that there is a good match between the two outputs with a very small error, in general. The modelling error of the training data set is smaller than the test data set. The mean absolute error (MAE) index is used to evaluate the modelling effects. For this model the MAE values of crankshaft speed, manifold pressure, manifold temperature and air fuel ratio are 0.0038, 0.0942, 0.0027 and 0.0029 respectively.

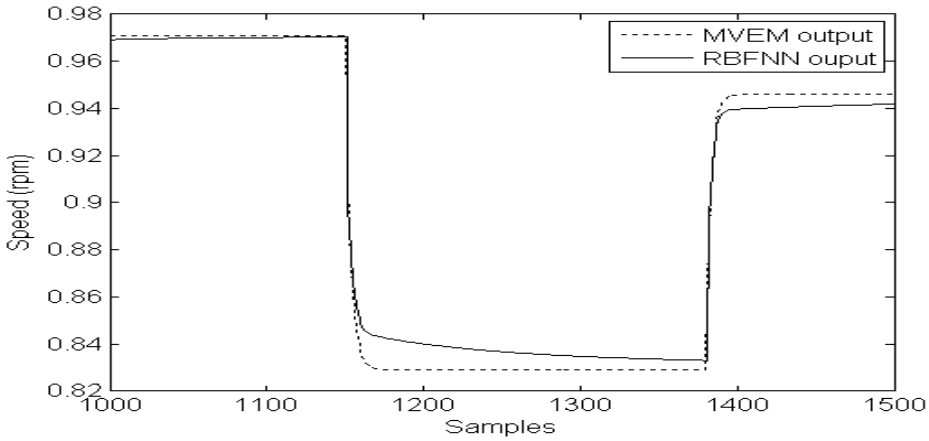



Figure 14 simulation results of the speed engine model output and the RBFNN output

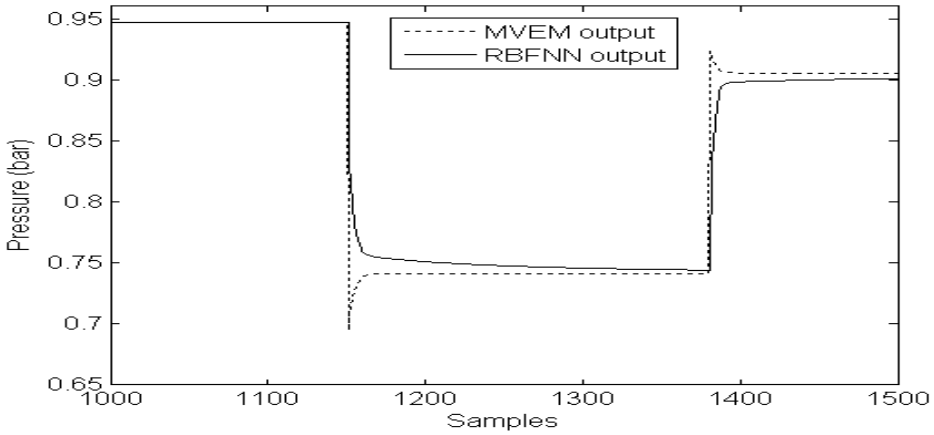



Figure 15 simulation results of the pressure engine model output and the RBFNN output

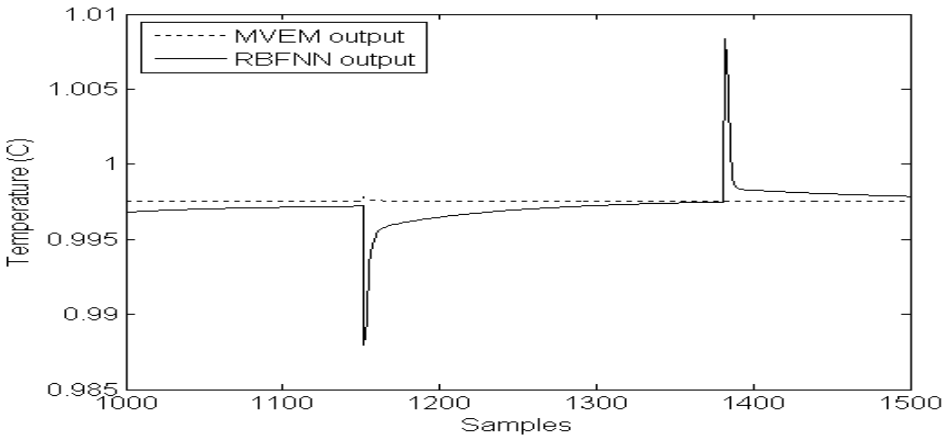



Figure 16 simulation results of the temperature engine model output and the RBFNN output

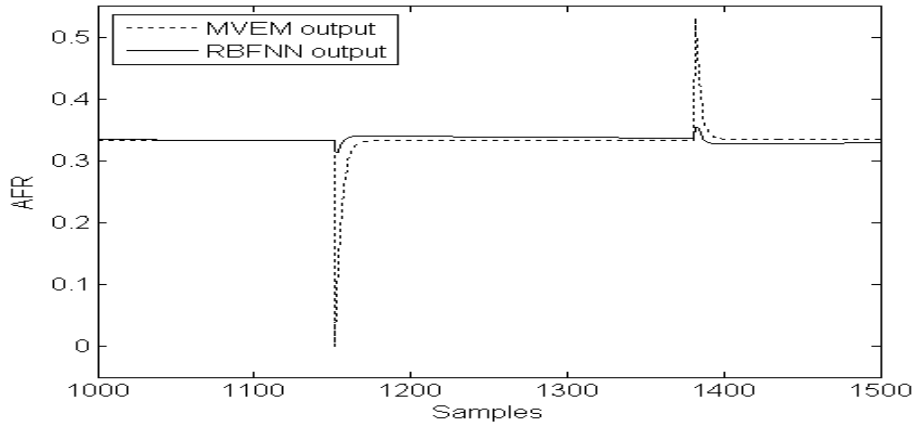



Figure 17 simulation results of the air fuel ratio engine model output and the RBFNN output

## 5. Fault detection and isolation

### 5.1 FDI strategy

The whole fault detection and isolation strategy is presented in Fig.18. Firstly, an independent neural network model is trained with engine data collected from the engine without any fault, which is called healthy condition. Then, the model is used parallel to the engine in on-line mode to predict engine output. The modelling error between the engine output and model prediction will be used as residual signal. Thus, if no fault occurs in the engine system, the residual is just modelling error caused by noise and model-plant mismatch. When any fault occurs, the engine output will be affected and will deviate from the nominal values, while the model prediction does not affected by the fault. So, the residual will have a significant deviation

from zero caused by faults. The proposed strategy can be applied to different systems to detect dynamic faults. In this study, the residual $\varepsilon$ is generated as the sum-squared filtered modelling error as follows,

$$e(k) = [y(k) - \hat{y}(k)] \tag{15}$$

$$e_{filtered}(k) = \mu e_{filtered}(k-1) + (1-\mu)e(k) \tag{16}$$

$$\varepsilon(k) = \sqrt{e_{filtered}^T e_{filtered}} \tag{17}$$

Fault isolation is achieved by an additional RBF neural network as a classifier. The modelling error vector (in this study it is a four-dimension vector) includes information of all faults, and different faults will affect each element in the modelling error vector in different ways. Thus, the modelling error vector will be used as the input to the classification RBF network. The RBF classifier is trained in the following way. Collect engine data for each of the faults occurring only, and then these data are fed into the classifier with the target of the output being all “0”s but the dedicated output being “1”. Thus, when the classifier is used to isolate faults in on-line mode, any output turns to “1” indicating that the associated fault occurs. By using this characteristic the faults can be isolated clearly if all possible faults have different characteristics and no more than two faults occurring at the same time. In this research totally six faults have been tested, so that the classifier has four inputs and seven outputs with six associated to the six faults and one for “no fault case”. It is noted that the classifier is a static network and has no dynamics of the residual involved.

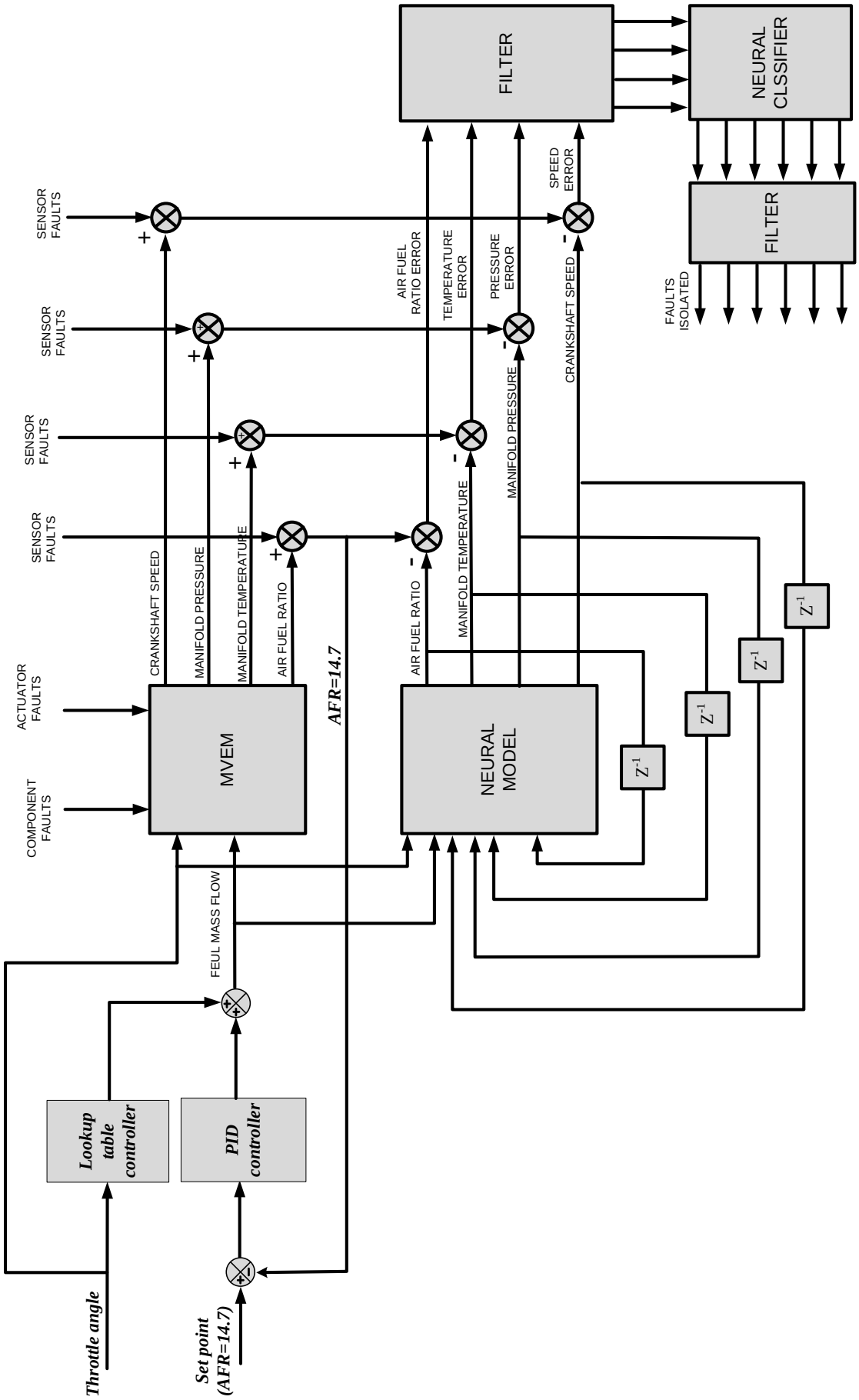



Figure 18 The information flow for the fault detection and isolation

**5.2 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.

5.2.1. Component fault

To collect the engine data subjected to the air leakage fault, equation (1) of the manifold pressure is modified to equation (18):

$$\dot{p}_i = \frac{RT_i}{V_i}\left(-\dot{m}_{at} + \dot{m}_{ap} + \dot{m}_{EGR} - \Delta l\right) \tag{18}$$

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 $\Delta l$ 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 $10\%$ of total air intake in the intake manifold. This fault occurs from the sample number 1450~ 1500 in the faulty data as shown in Fig.19, and was simulated by changing the Simulink model of the MVEM.

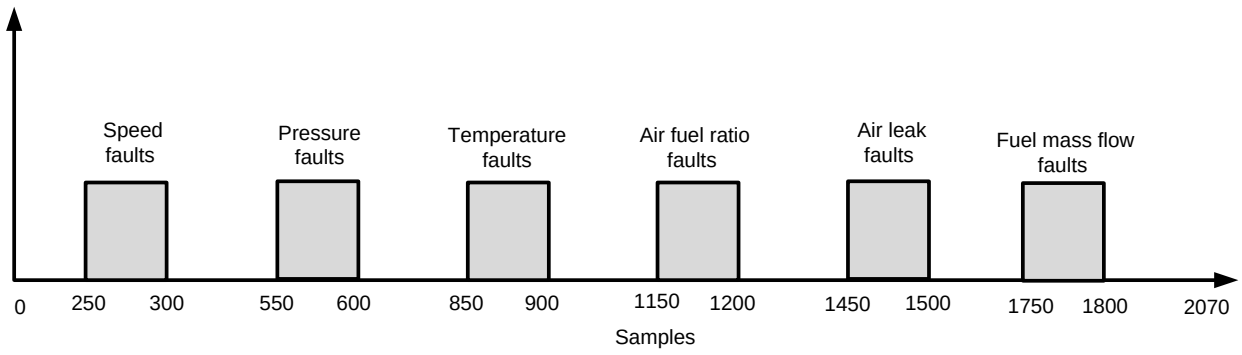



Fig. 19 Time distribution of the simulated faults

5.2.2 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 15% of the total fuel mass flow rate between the sample number 1750 and 1800 as shown in Fig.19. This fault is also simulated by changing the Simulink model of the MVEM.

5.2.3 Sensor fault

The four sensor faults considered are (10, 15, 10 & 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 250 to 300, 550 to 600, 850 to 900 and 1150 to 1200 respectively. The faulty data for the sensors is generated using multiplying factors (MFs) of 1.1, 1.15 or 1.2 for the above over-reading faults respectively, see Figure 19.

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 3. The sample time is chosen as 0.02 sec.

Table 3 The 6 Faults States and Multiplying Factors

| No | Fault Name | multiplying factors (MFs) |
|---|---|---|
| 1 | Air Leak 10% | |
| 2 | Injected fuel mass flow 15% | |
| 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 | 1.2 |

**5.3 Fault detection**

Fig.20 shows the flow chart of fault detection and isolation. Firstly, the random amplitude sequence for throttle angle in the proper range and the output of the PID controller are fed into the MVEM. The collected four engine output together with the two input 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 set of RAS fed into the MVEM, the model prediction error and the filtered residual are generated for fault detection. With the six faults simulated from samples 250 to 1800 as shown in Fig.19, the generated model prediction error are shown in Fig.21-24. The first model prediction error of air fuel ration is shown in Fig.21. The second, third and fourth for air manifold temperature, pressure and engine speed are showed in Figs.22 ~24 respectively. In these figures the samples 0 to 250 are data without faults. Including them is to show the prediction error is under the selected threshold in "no fault case". However, it is not very clear in Fig.21-24 that the prediction error is over the threshold when a fault occurs, due to the noise.

After a low-pass filter is used the filtered prediction errors are shown in Fig.25 (b, c, d & e). 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 figures. Finally the residual as defined in (17) is computed and displayed in Fig.25, f.

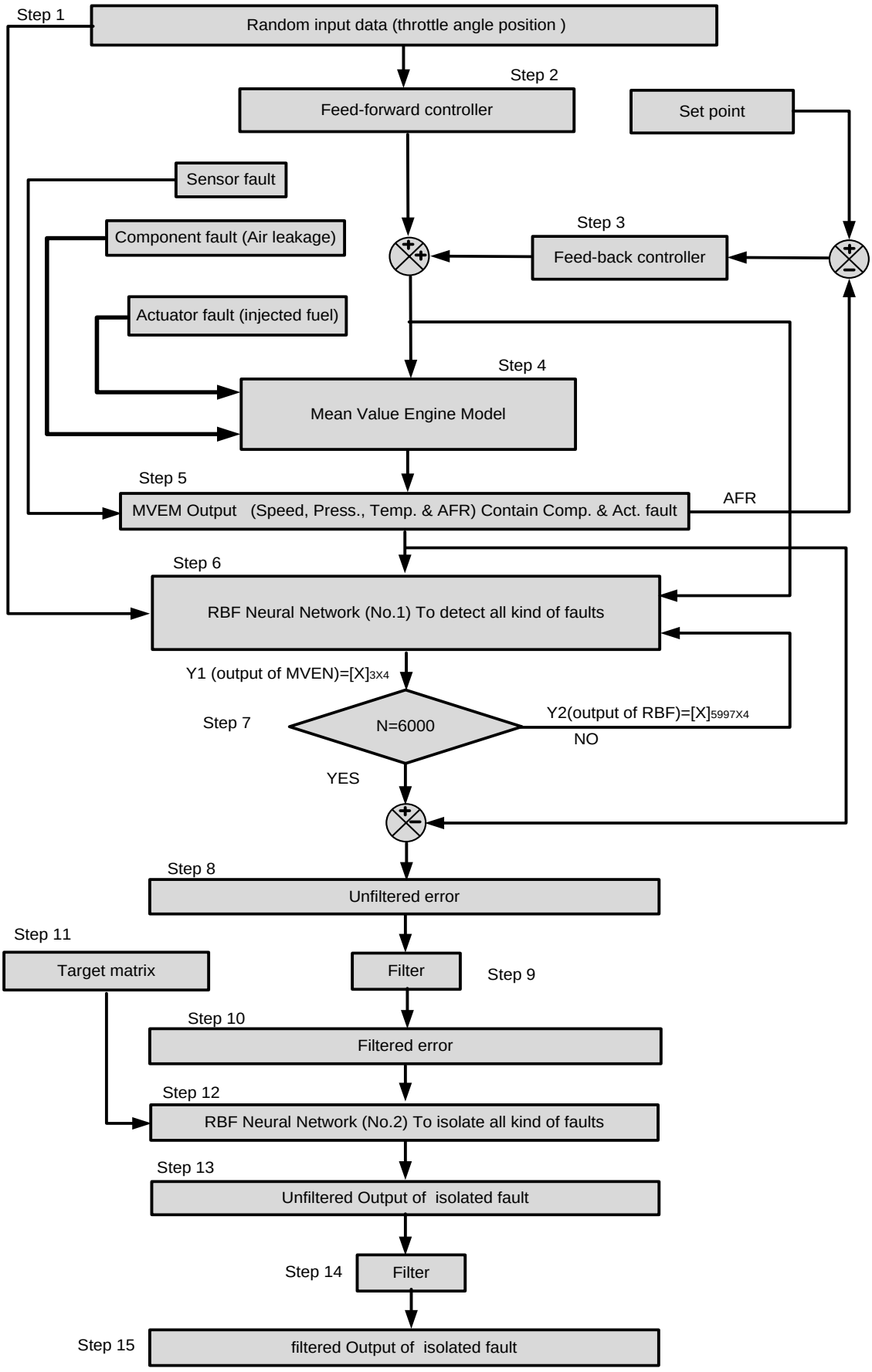
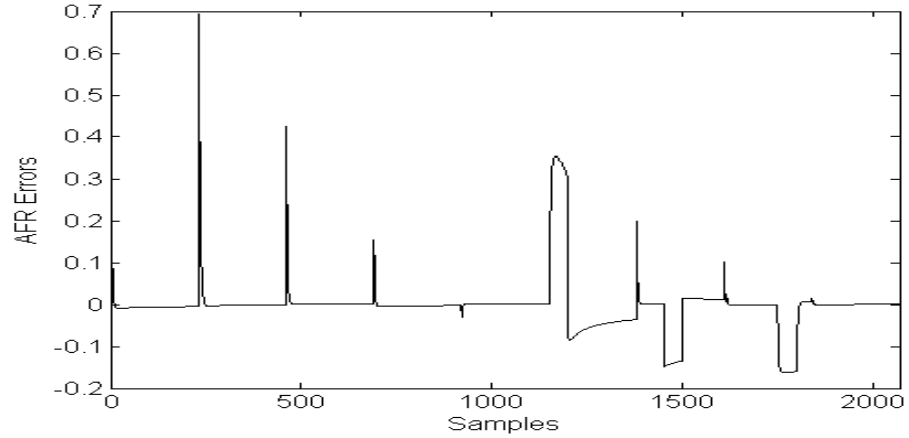
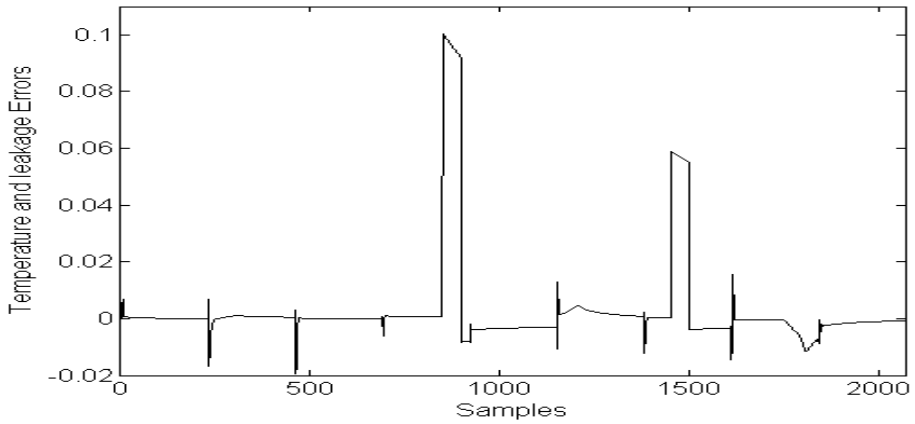


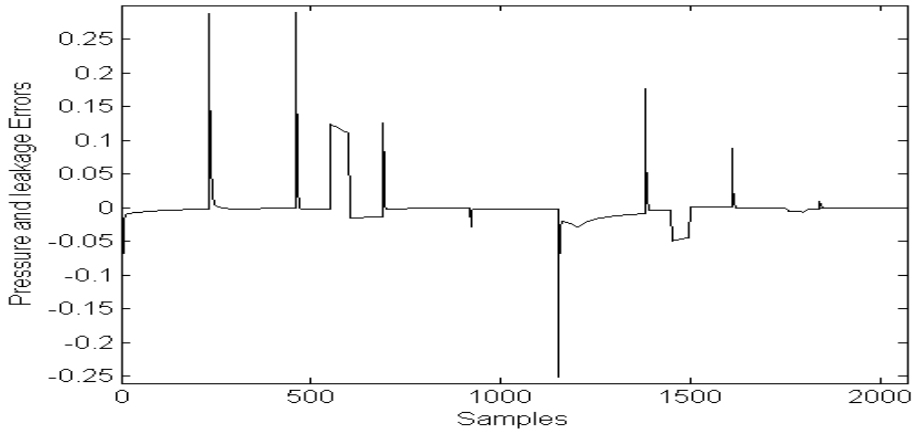

Fig.20 Flow chart of fault detection and isolation



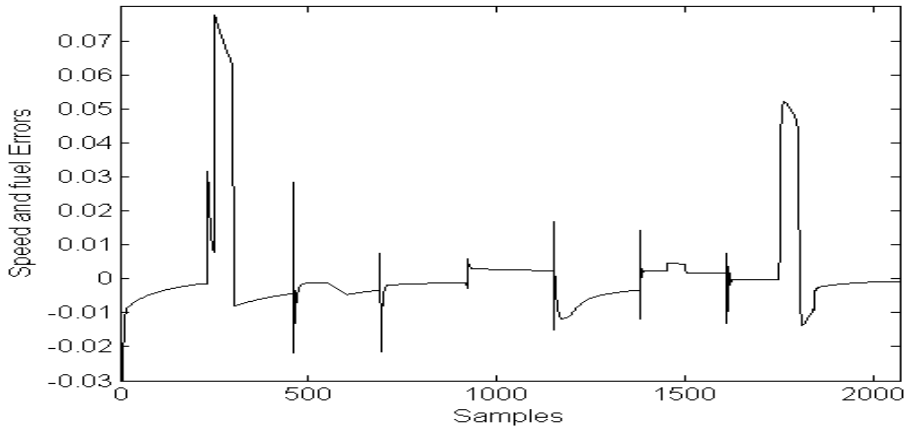

Figures 21 Model prediction error of air fuel ratio



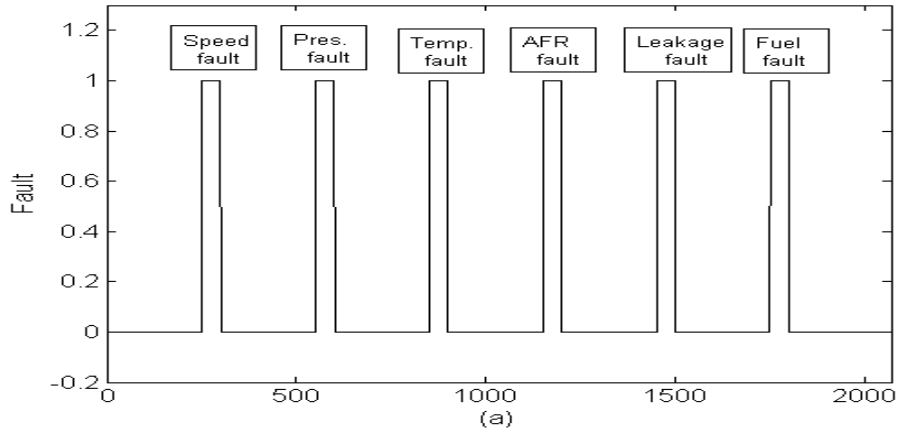

Figures 22 Model prediction error of air manifold temperature and leakage

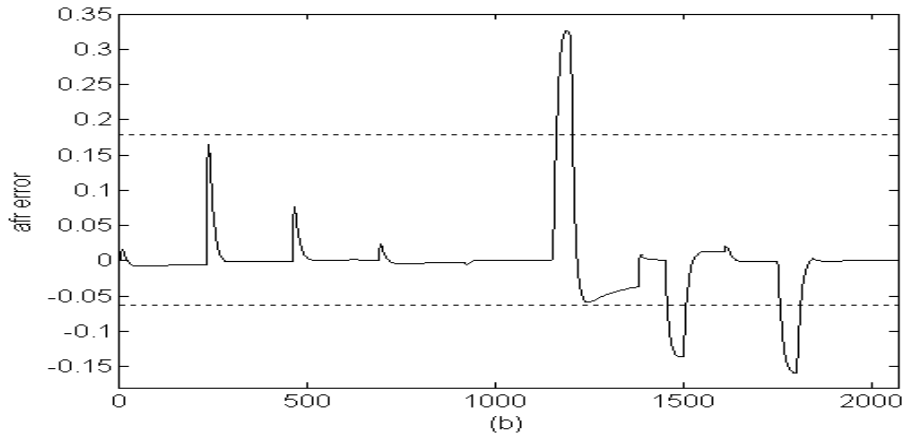


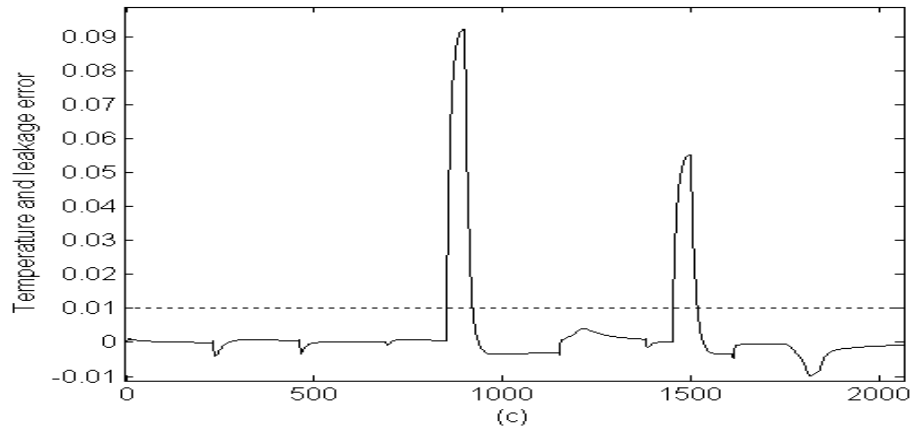

Figures 23 Model prediction error of air manifold pressure and leakage

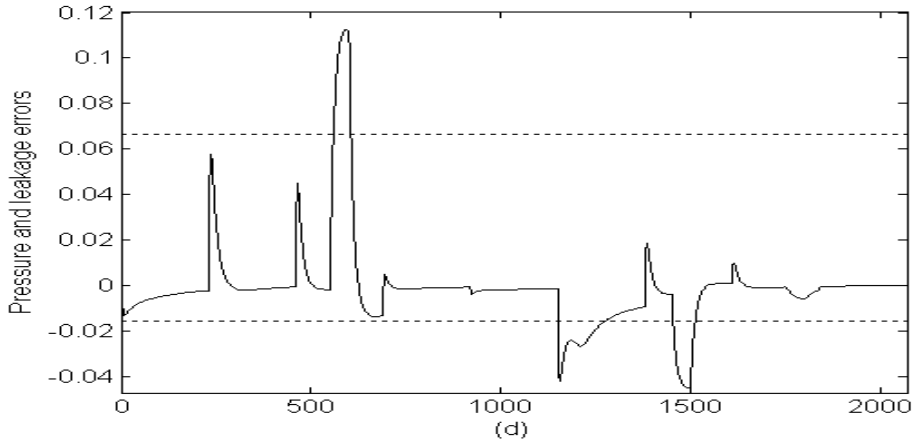


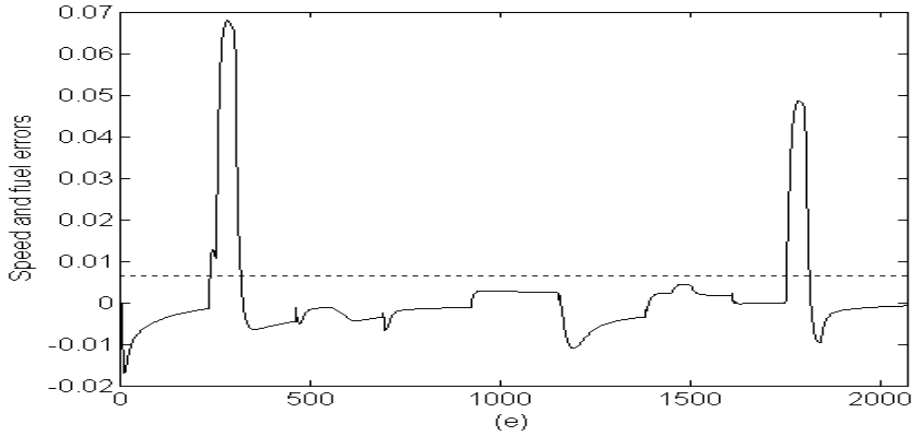

Figures 24 Model prediction error of engine speed and fuel




Temperature and leakage error
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
0
500
1000
1500
2000
(c)


Pressure and leakage errors
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
0
500
1000
1500
2000
(d)


Speed and fuel errors
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
0
500
1000
1500
2000
(e)

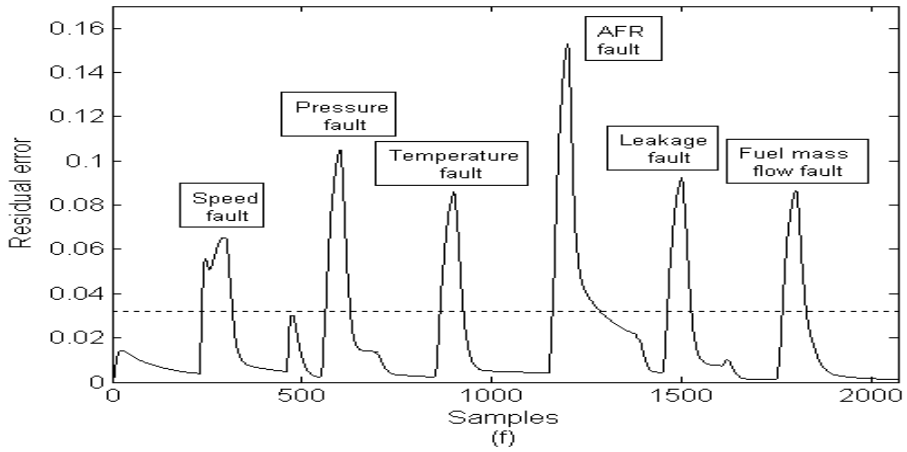



Fig. 25 (a)simulated faults, (b), (c) , (d) and (e) Filtered model prediction error of air fuel ratio, air manifold temperature, air manifold pressure and engine speed , (f) Filtered residual

**5.4 Fault isolation**

As have been seen in Fig.29, 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 modelling errors. It can be concluded from Fig.25 that If only 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 third model output (pressure) go over the thresholds while the other two outputs remain under the thresholds, then it must be the air fuel ratio 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 modelling 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 2070 data samples were collected with first 1770 without fault and each set of 300 samples with one of the six faults. These data are fed into the RBF model and the generated four modelling errors are fed into the RBF classifier, with the targets given as described above. After training, the classifier is tested with the similar arrangement of data, totally 2070 data samples. The first 250 samples are fault-free, followed by six data sets with each having 50 samples and having a single fault. Between any two of these six faulty data set insert 250 fault-free samples and the final 270 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 3.

Table 4 Data samples and fault types

| Data samples | Fault types |
|---|---|
| 1 ~ 250 | No fault |
| 251 ~ 300 | Speed sensor fault |
| 301 ~ 550 | No fault |
| 551 ~ 600 | Pressure sensor fault |
| 601 ~ 850 | No fault |
| 851 ~ 900 | Temperature sensor fault |
| 901 ~ 1150 | No fault |
| 1151 ~ 1200 | Air fuel ratio fault |
| 1201 ~ 1450 | No fault |
| 1451 ~ 1500 | Air leak fault |
| 1501 ~ 1750 | No fault |
| 1751~1800 | Fuel injector fault |
| 1801~2070 | No fault |

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 $\mu$= 1.0, $w(0)=1.0\times10^{-6}\times U_{(nh\times6),}$ $P(0)=1.0\times10^{5}\times I_{(nh).}$ The number of the hidden layer nodes was tried several numbers and the one giving minimum training error was chosen and was 250. Figs. 26-37 show the test result before and after filtering. The isolation thresholds are chosen as 0.1 for all cases.

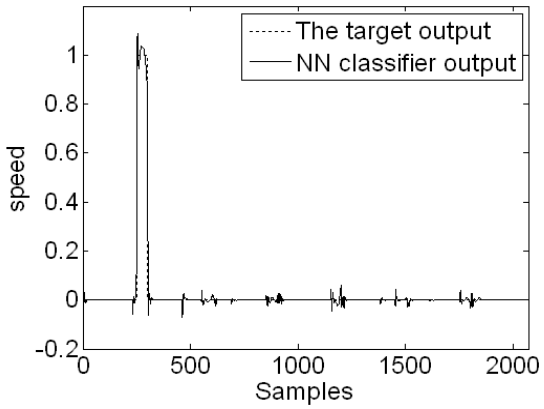



Fig. 26 First output of fault classifier

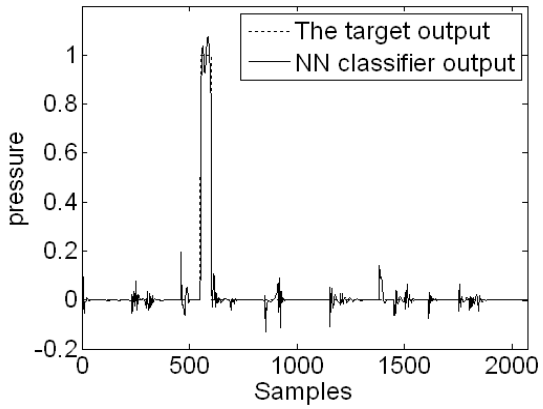



Fig. 27 Second output of fault classifier

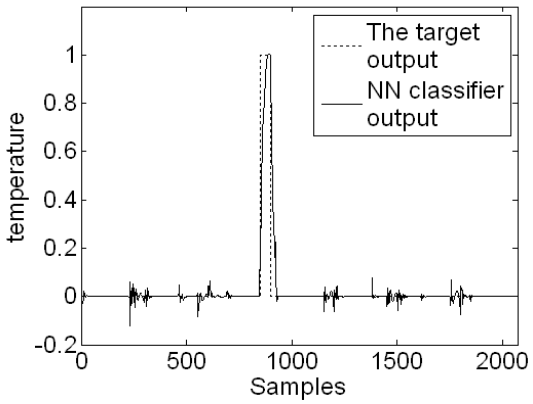



Fig. 28 Third output of fault classifier

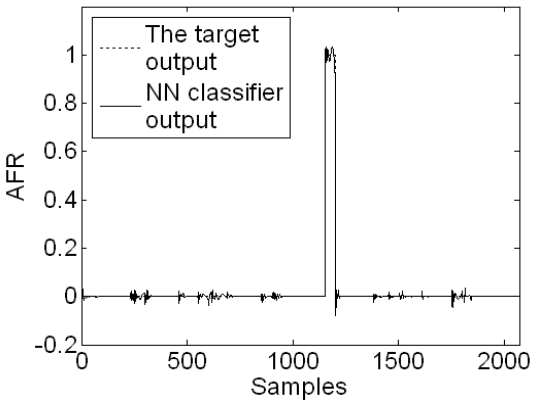



Fig. 29 Forth output of fault classifier

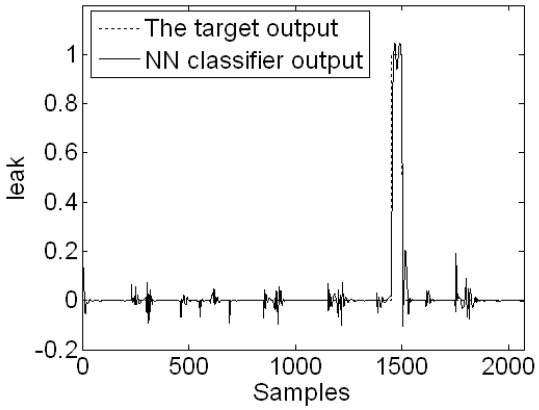



Fig. 30 Fifth output of fault classifier

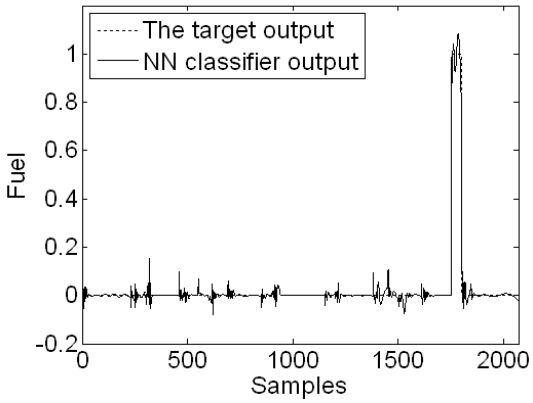



Fig. 31 Sixth output of fault classifier

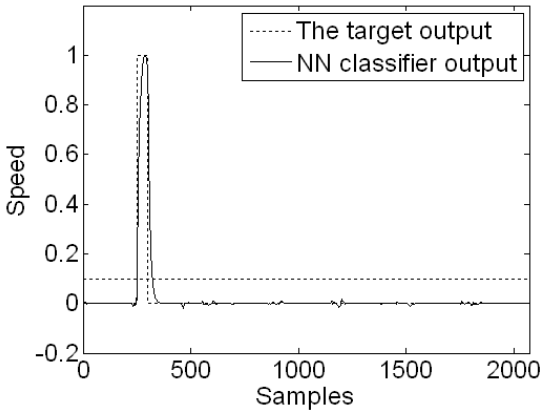



Fig. 32 Filtered first output of fault classifier

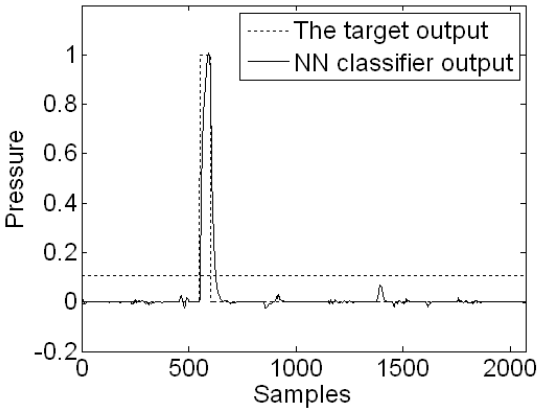



Fig. 33 Filtered second output of fault classifier

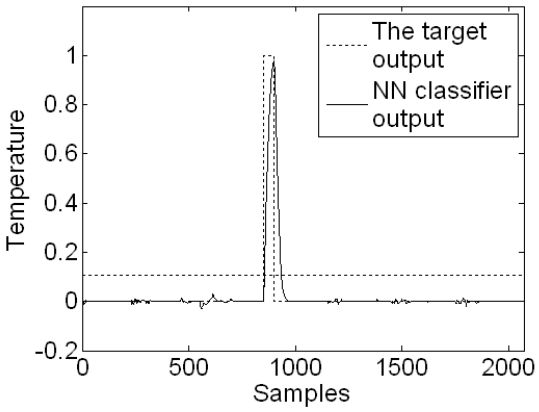



Fig. 34 Filtered third output of fault classifier

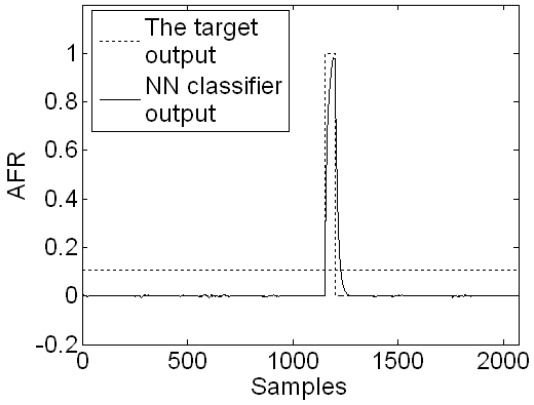



Fig. 35 Filtered forth output of fault classifier

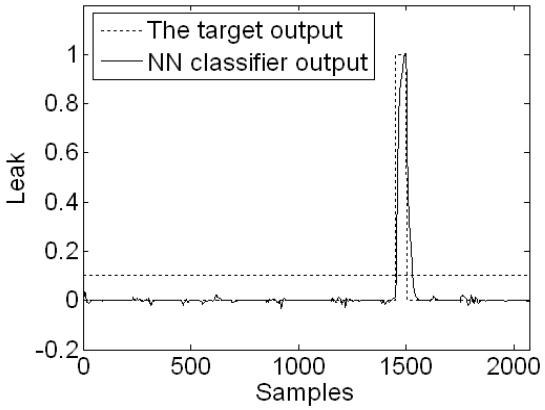



Fig. 36 Filtered fifth output of fault classifier

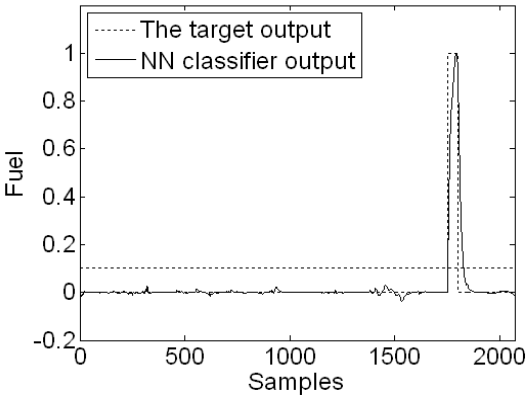



Fig. 37 Filtered sixth output of fault classifier

**6. Discussion of simulation results**

We can summarize the results in the following points:

6.1. 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 Figures 14, 15, it can be seen that the mean absolute error between the engine speed output and the RBFNN is very small; however we can see a mismatch in Figures 16 and 17, the mean absolute errors between the three outputs are shown in table 5.

Table 5. Error values between the engine model outputs and the RBF neural network outputs

| Outputs | Mean Absolute Errors |
|---|---|
| Crankshaft speed | 0.0038 |
| Pressure | 0.0942 |
| Temperature | 0.0027 |
| Air fuel ratio | 0.029 |

6. 2. Detection of the Sensor, Component and Actuator Faults.

The Figures 21 ~ 24 show the test results of the fault detection of air fuel ratio, temperature/air leak, pressure/air leak, and speed/fuel injection respectively before filtering with 15 hidden nodes. While

the Figure 25 (b, c, d and e) after filtering operation. It can be seen from figures after filtering operation that all kind of faults were detected clearly. And the error values were between -0.05 and 0.02 except the samples in which faults occur. The detection thresholds were chosen as 0.008 for crankshaft speed / fuel injection, (- 0.018/+0.065) for manifold pressure/air leak, 0.01 for manifold temperature/air leak and (-0.06/0.18) for air fuel ratio.

### 6. 3. 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 figures of the results, we found 250 hidden nodes after filtering operation were the best case and the simulated faults can be clearly isolated (see figures 32~37). The isolation thresholds are chosen as 0.1 in case 250 hidden nodes.

## 7. Conclusions

In this research, the MVEM under close loop control including feed-forward and feed-back controllers was used. Look up table was 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 modelling 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 figures of simulation results, it can be seen that the methods were

able to detect and isolate the faults.

## References


Adnan hamad, Yu. D. L, Gomm, J.B. and Sangha, Mahavir. (2010), 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* ,pp.232-237 , ISBN978-0-9555293-6-8,Birmingham, UK.

Balluchi, A., Benvenuti, L., Di Benedetto, MD., Pinello, C., SangiovanninVincentelli, AL. (2000), Automotive engine control and hybrid systems: challenges and opportunities. *Proceedings of the IEEE*, 88(7), 888-912.

Beham, M., Yu, D.L. (2004). Modelling a variable valve timing spark ignition engine using different neural networks *Proc. Inst. Mech. Eng. Part D: Journal of Automotive Engineering*, Vol.218, No.10, pp. 1159-1172,

Blanke, M. and Patton, R.J. (1995). Industrial actuator benchmark for fault detection and isolation, *Control Engineering Practice*, Vol.3, No. 12, pp.1727-17307.

Capriglione, D., Liguori, C., Pianese, C. and Pietrosanto, A. (2003). On-line sensor fault detection, isolation, and accommodation in automotive engines, *IEEE Trans. on instrumentation and measurement*, Vol. 52, No. 4, pp.1182-1189.

Capriglione, D., Liguori, C., Pianese, C. and Pietrosanto, A. (2004), Analytical redundancy for sensor fault isolation and accommodation in public transportation vehicles, *IEEE Trans. on Instrumentation and Measurement*, Vol. 53, No.4, pp. 993-999.

Capriglione, D., Liguori, C. and Pietrosanto, A. (2007), Real – Time Implementation of IFDIA Scheme in Automotive Systems, *IEEE Trans. on Instrumentation and Measurement*, Vol. 56, No.3, pp. 824-830.

De Vegte, John Van, (1994), Feedback Control System. *Prentice-Hall international, inc*

Gertler J., Costin M., Fang X., Hira R., Kowalczuk Z. and Luo Q., (1993), Model based on board fault detection and diagnosis for automotive engines, *Control Engineering Practice*, Vol. 1, No.1, pp 3-17.

Gertler J., Costin M., Fang X., Kowalczuk Z., Kunwer M., and Monajemy R., (1995), Model based diagnosis for automotive engines – algorithm development and testing on a production vehicle, *IEEE Trans. on Control Systems Technology*, Vol. 3, No.1, pp 61-69.

Hendricks, E., Engler, D. and Fam, M. (2000). A Generic Mean Value Engine Model for Spark Ignition Engines. *Proceedings of 41st Simulation Conference*, Denmark, DTU Lyngby.

Isermann, R. (2005). Model-based fault-detection and diagnosis status and applications, *Annual Reviews in Control*, Vol.29, pp.71–85.

Joseph J. Distefano, III, Allen R. Stubberud and Ivan J. Williams. (1990). McGRAW-Hill, INC

Kimmich, F., Schwarte, A. and Isermann, F. (2005). Fault detection for modern Diesel engines using signal- and process model-based methods, *Control Engineering Practice*, Vol.13, pp.189–203,

Ogata, Katsuhiko, (1997). Modern control engineering. *Prentice-Hall international, inc*

S. W. Wang, D. L. Yu, J. B. Gomm, G. F. Page and S. S. Douglas (2006). Adaptive neural network model based predictive control for air- fuel ratio of SI engines, *Engineering Applications of Artificial Intelligence*, Vol.19, pp.189-200.

Victor Hillier and Peter Coombes (2004). *Fundamentals of motor vechicle technology*. Nelson thornes Ltd.

Yu, T and Rizzoni, G. (1991). A processor architecture for real-time fault detection and isolation in electronically controlled diesel engines, *Proceedings of Vehicular Technology Conference*, Seattle, USA.

Zhai Y. J. and Yu. D. L (2007). Radial Basis Function Based Feeddback Control for Air Fuel Ratio of Spark Ignition. *Proc. IMACE Joural of Automobile Engineering*, Vol. 222, pp. 415-428.