FAULT DETECTION AND IDENTIFICATION OF AUTOMOTIVE ENGINES USING NEURAL NETWORKS

M. S. Sangha, J. B. Gomm, D. L. Yu, G. F. Page

Control Systems Research Group, School of Engineering,
Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK
E-mail: j.b.gomm@livjm.ac.uk

Abstract: Fault detection and isolation (FDI) in dynamic data from an automotive engine air path using artificial neural networks is investigated. A generic SI mean value engine model is used for experimentation. Several faults are considered, including leakage, EGR valve and sensor faults, with different fault intensities. RBF neural networks are trained to detect and diagnose the faults, and also to indicate fault size, by recognising the different fault patterns occurring in the dynamic data. Three dynamic cases of fault occurrence are considered with increasing generality of engine operation. The approach is shown to be successful in each case. *Copyright © 2005 IFAC*

Keywords: fault diagnosis and isolation, radial basis function networks, classification, artificial intelligence, neural networks, engine systems.

1. INTRODUCTION

Electronic engine control was introduced in the1970s, mainly as a result of two government requirements. The first was legislation to reduce and regulate automobile exhaust emissions and the second was the desire to improve the national average fuel economy. Evidently, the same factors are the main reasons for more recent legislation for on-board engine diagnostic capability. The environmental Protection Agency (EPA), as well as the California Air Resource Board (CARB) mandated "On Board Diagnosis II" (OBD-II) for all light duty vehicles sold in American fleet starting from 1996 (*Tan and Saif, 2000*). All diesel cars sold in the EU must have an OBD system from year 2003 (*Nyberg and Stutte, 2004*). The same is expected for gasoline cars and heavy trucks, etc. OBD-II requires continuous monitoring and fault detection capability for all vehicle components whose failures can result in emission levels beyond 1.5 times of the Federal Test Procedure (FTP) standards.

A variety of diagnosis methods have been proposed under the umbrella of model-based techniques. The feature of all these techniques is that some form of mathematical knowledge of the process of interest along with inputs and outputs are used to generate superfluous information about that process. This redundant information is then used in a diagnostic process to arrive at decisions regarding fault or no-fault conditions. Structured hypothesis based on statistical hypothesis tests (SHT), observer based nonlinear estimation (*Yong-Wha, 1998*), real time supervision using production sensors and additional sensors installed (*Nyberg and Stutte, 2004)* and use of artificial intelligence, i.e. fuzzy logic (*Laukonen et al., 1995*) and neural networks (*Gomm et al., 2000*), are examples of the popular FDI systems. In this paper neural networks are used for fault diagnosis in the air path of an automotive engine. There are many neural network architectures available for FDI, which can broadly be divided into supervised and unsupervised networks. Radial Basis Function (RBF) and Multilayer Perceptron (MLP) are examples of supervised neural networks whereas ART2 and Kohonen network of unsupervised networks (*Gomm, et al. 2000*). Supervised networks have been shown to exhibit better classification capabilities than unsupervised networks in FDI (*Sorsa and Koivo,* *1993*). In this research, four different faults with four different levels of intensities are considered i.e. air leakage in the intake-manifold, Exhaust Gas Recycle (EGR) valve stuck in different positions, intake-manifold pressure and temperature sensor faults. These four faults are similar to those studied in other research (e.g., *Nyberg and Stutte, 2004*). There are 17 states considered in all including no fault state. The transient data for each different state is collected for equally distributed throttle angle inputs ranging from 20^o to 40^o and RBF networks are used for fault pattern classification.

2. MEAN VALUE ENGINE MODEL (MVEM)

The engine models are developed based on the engine's physical characteristics and some steady-state engine data (*Cook and Powell, 1993*). Such models consist of non-linear differential equations, table look-ups, regression-based functions and empirical formulae. It is important to note that the models derived from the empirical data or physical phenomena are generally constructed under idealized situations, as such, they may not completely represent the practical dynamics or non-linear behaviour of engines.

All IC engines contain significant nonlinearities, which dominate their dynamic behaviour. The MVEM is a fairly good approximation of medium speed IC engine dynamics. This model includes the latest results and efficiency enhancement system such as Exhaust Gas Recycle (EGR) unit. A detailed derivation of all the sub-models of the MVEM can be found in *Hendricks, et al. (2000, 1996* and *1993).* Only some main mathematical details of relevant sub-models are discussed in this paper because the main focus of this research is on Fault Detection & Isolation (FDI) and not on the MVEM simulation. The block diagram of MVEM is shown in fig 1.

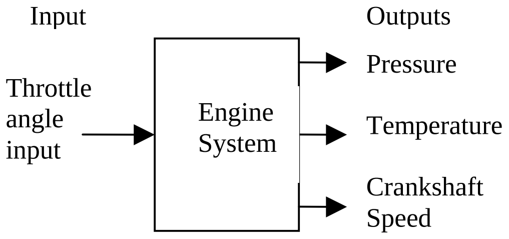

Fig 1: Engine block diagram

2.1 Manifold Filling Dynamics:

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

$$\dot{p}_i = \frac{\kappa R}{V_i}\left(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR}\right) \tag{1}$$

The only problem with adiabatic assumption is that the intake manifold temperature must be known accurately and instantaneously (*Hendricks, 2001*), whereas the traditional temperature transducers have a time constant of three seconds. Using the law of energy conservation a state equation which describes the time development of the intake manifold temperature can be given as:

$$\dot{T}_i = \frac{RT_i}{p_iV_i}\left[\begin{array}{l} -\dot{m}_{ap}(\kappa-1)T_i + \dot{m}_{at}(\kappa T_a - T_i) \\ +\dot{m}_{EGR}(\kappa T_{EGR} - T_i) \end{array}\right] \tag{2}$$

2.2 Crank Shaft Speed State Equation

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

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

where I is the scaled moment of inertia of the engine and its load and the mean injection/torque time delay has been taken into account with variable $\Delta\tau_d$.

2.3 Simulation of Adiabatic System

The engine system is implemented in a block diagram in Simulink and has throttle angle as the only input and manifold temperature, pressure and crankshaft speed as the outputs. The simulation is run for 6 seconds for every fault with solver options set to variable step for ODE45 in MATLAB.

3. FAULT-SIMULATIONS

Leakage in the intake manifold fault and EGR valve stuck up in different positions fault are simulated in the Simulink model of the engine where as the sensor faults are calculated from normal temperature and pressure values of the intake manifold. All the four faults with four different intensities are simulated one at a time in the model, and all the three outputs i.e. pressure; temperature and crankshaft speed are recorded for different faults.

3.1 No Fault

For no fault situation, EGR is assumed $1/6^{th}$ (16.67%) of the total air mass flow in the intake manifold. It is also assumed that all the sensors are working well and there is no leakage in the intake manifold. The simulation is run for 6 seconds for no fault condition.

3.2 Air Leakage Fault

Equation (1) can be modified as

$$\dot{P}_i = \frac{\kappa R}{V_i}\left(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR} - k\right) \tag{4}$$

Where constant k is added in the model to increase 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 (5%, 10%, 15%, and 20%) of the total air intake in the intake manifold.

3.3 EGR Valve Faults

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 $\dot{m}_{EGR}$ for different fault intensities are regulated as 0%, 25%, 50%, 75% and 100% of the total EGR air mass flow. Where 0% EGR air mass flow corresponds to the EGR valve stuck in hundred percent-closed position and 100% corresponds to full EGR air mass flow, i.e. no fault condition.

3.4 Temperature/Pressure Sensor Faults

Temperature and pressure sensor faults are considered in four different intensities: Sensors over-reading 20% or 10% and sensors under-reading 10% or 20% of the normal value The fault data for the sensors is generated using multiplying factors (MFs) of 1.2, 1.1, 0.9 and 0.8 for over reading 20%, 10% and under reading 10%, 20% respectively. Fault data for throttle angles between 20^{o} and 40^{o} are generated for all the fault conditions including no fault condition. All the 17 fault states with MFs are given in table 1.

Table 1: All 17 fault states and MFs

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

4. NORMALISATION OF DATA

Data normalisation is necessary before inputting it through a Neural Network for learning so that higher numerical value data may not dominate the learning. More over the graphical data analysis will be difficult due to huge difference in the numerical values of data. One way to normalise data is to find deviation from the normal steady state as given in equation (5).

$$\text{Deviation} = (x - x_{ss})/x_{ss} \qquad (5)$$

Considering steady state values of all the three outputs as normal at 30^{o} throttle angle input for no fault condition, the deviation is calculated using above formula along with the use of a proper multiplier. Throttle angle input is also normalised and is used as an input to the neural network for training. The four neural network inputs are graphically shown in fig 2, 3, 4 and 5.

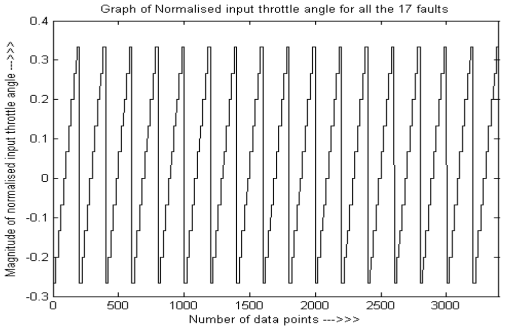

Fig 2: Normalised throttle angle input graph for all the 17 faults for case 1 (section 6.1)

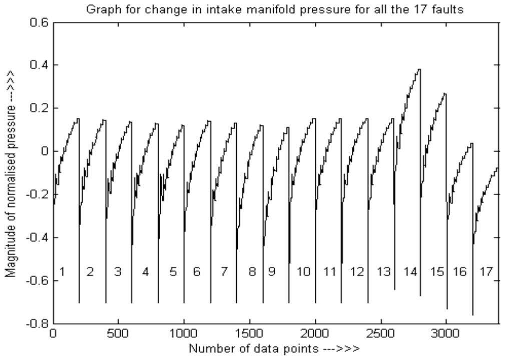

Fig 3: Normalised manifold pressure response for all the 17 faults for case 1 (section 6.1)

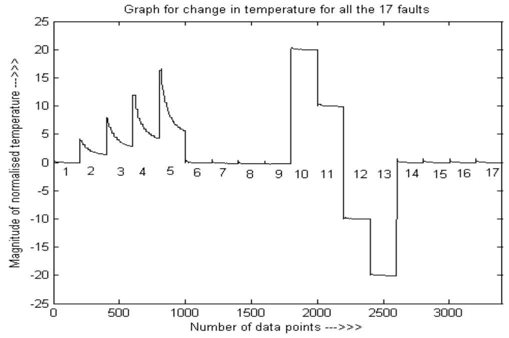

Fig 4: Normalised manifold temperature response for all the 17 faults for case 1 (section 6.1)

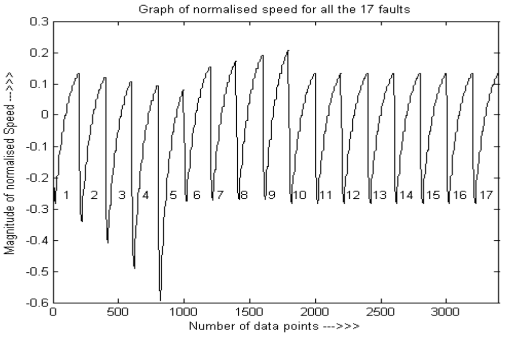

Fig 5: Normalised crankshaft speed response for all the 17 faults for case 1 (section 6.1)

5. NEURAL NETWORK ARCHITECTURE

The architecture of an RBF network consists of three layers; input, hidden and output. The input layer simply receives the network inputs i.e. fault signals (x) and passes them to each node in hidden layer. The hidden layer consists of n_h nodes that process the input vector. Each node in the hidden layer contains individual centre vector c_i and a width ρ_i. The Euclidean distance between the input and the centre vectors is calculated:

$$z_i = \|x - c_i\| = \sqrt{(x_1 - c_{i1})^2 + ... + (x_n - c_{in})^2} \quad (6)$$

where $i = 1,...,n_h$, and passed through a non-linear basis function to produce hidden node outputs ϕ_i. Several choices of basis function are available, i.e. thin plate spline, Gaussian function, etc. Gaussian basis functions provide a local excitation of the node with an output ϕ_i near zero for inputs far from the centre. This is especially suitable for fault diagnosis applications and is used in this work. The Gaussian basis function is defined as

$$\phi_i = \exp\left[-(z_i/\rho_i)^2\right], \quad \rho_i > 0 \quad (7)$$

Finally the network outputs are computed as a linear weighted sum of the hidden node outputs:

$$y = W^T \phi \quad (8)$$

where y is the output vector of m outputs, W is the output layer weight matrix with element w_{ij} connecting hidden node i to output j, and ϕ is a vector containing the hidden node outputs.

RBF networks are trained using the orthogonal least squares algorithm (Chen et al., 1991), which builds the network up by one hidden node at a time to minimise the output error. The same width value is used in each hidden node and is chosen by trial and error for a small network size with acceptable error level. MLP networks were also trained but they take much longer time to train and their performance was poorer than RBF networks (Sangha et al. 2004).

6. TRAINING & TESTING OF NNs

Three cases are considered with increasing generality of engine operation.

6.1 Case 1: Engine accelerates or retards from mean speed and a fault occurs:

In order to train NN dynamically, throttle angle input of 20^o, 22^o, 24^o, …, 40^o is applied for each fault condition, keeping all the initial conditions set to 30^o throttle angle. It is assumed that the engine is initially running for 30^o throttle angle before accelerating or retarding the speed i.e. increasing or reducing the throttle angle. 30^o is chosen as the initial condition because it is the mean value of the selected operation range of the engine. . Each fault is simulated for 6 seconds (20 data points) for each throttle angle. All the data is collected and normalised. The RBF is trained from this data, for the whole range of throttle angles from 20^o to 40^o with an error thresh hold of 300 and spread constant of 0.5. The throttle angle is also used as an input along with pressure, temperature and speed as shown in fig 6.

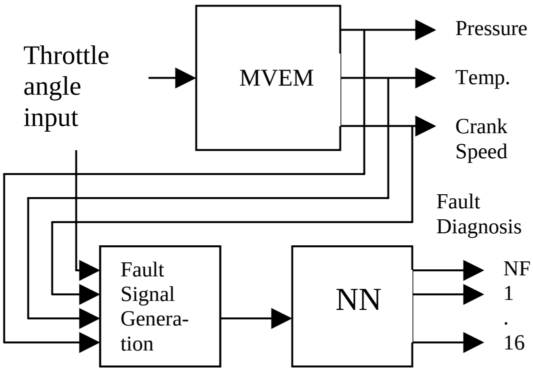

Fig 6: NN scheme for fault diagnosis

It creates a bulky input training-data-matrix of 4x3400. Training takes as long as 15 minutes on a P4 computer and forms a network of 191 neurons. Finally the trained NN is tested for different throttle angle inputs for all the 17 faults. For example data generated for 26^o throttle angle input is used for testing the NN. It is a case when the engine is retarding from 30^o throttle to 26^o throttle angle input and a fault occurs. The artificial NN is found very well able to classify all the faults correctly.

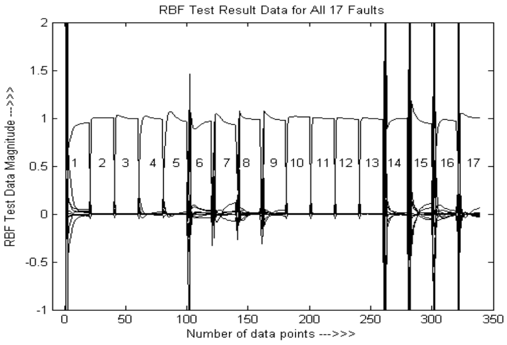

Fig 7: RBF Test Results for case 1

We find misclassification less than 0.3% (Fig 7). It is observed that the misclassification occurs at the beginning of the change of one fault to the other. Thus by ignoring the first three data sets of each fault test result, misclassification can be reduced to 0% (Fig. 8). In other words the initial three data sets, out of 20 data sets for each fault, are not used in making a decision regarding classification of each fault.

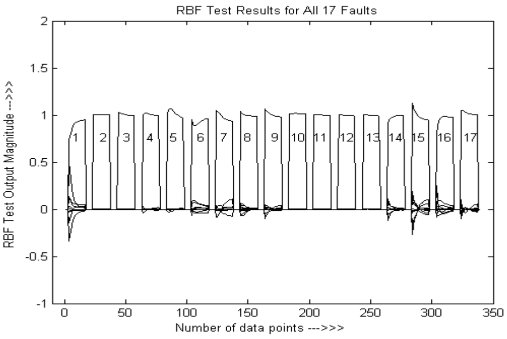

Fig 8: RBF Test Results (Improved) case 1

6.2 Case 2: Engine runs at constant speed at least for 6 seconds and a fault occurs:

The output response time for MVEM is about 6 seconds. If we are able to get a 6 seconds sample data of all the output transients (20 points sampled at 0.3 seconds) then it can be used to classify a fault or no fault condition in the engine.

For this part of the research, output data for all the three outputs, is collected for different throttle angle inputs and the engine is initialised for the same throttle angle input. Thus it is assumed that the engine is running at fixed throttle angle for 6 seconds. The training data is collected for 11 different throttle angles ranging from 20^o, 22^o, 24^o, …, 40^o for all the 17 different faults. The training data sizes 4x3740 and target data matrix 17x3740. An RBF NN is trained for an error threshold of 300 and spread constant of 0.2 and it results in a network of 130 neurons. The trained network is then tested for different throttle angle input data and 0% misclassification is found, which is an ideal result (Fig 9):

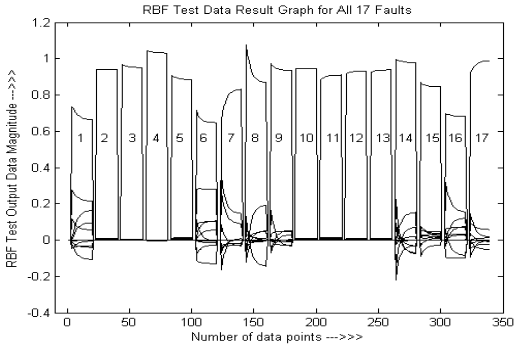

Fig 9: RBF Test Results for case 2

6.3 Case 3: Engine accelerates or retards from any initial speed and fault occurs:

This is the most general case when engine initial conditions are any value and engine accelerates or retards to any value and a fault occurs at the same time. The previous two cases are subsets of this case. The engine is initialised to different throttle angles for 6 seconds and then accelerated and retarded to other throttle angles as shown in Table 2.

Table 2: Training data for case 3

Initial Throttle Angle	Engine Speed Accelerating to throttle angle	Engine Speed Retarding to throttle angle
22	26,30,34,38	Nil
26	30,34,38	22
30	34,38	22,26
34	38	22,26,30
38	Nil	22,26,30,34

One training data set out of the above five data subsets is made to train one NN. It becomes very bulky if 20 data points are collected for each six second input time for all the 17 faults. It is important to collect at least 20 data points to represent the dynamic response of the engine properly. The sizes of the two input matrices for RBF are 17x8500 and 4x8500. This large amount of data caused an out of memory message from MATLAB during network training. An alternative way is to train five different NNs for five different subsets of data and then use them in parallel for testing purpose. This method is also tested in this research. The difficulty with this method is that it does not interpolate the unseen test data to an acceptable level.

In order to reduce the training data volume, all the four faults are considered with two levels of intensities instead of four as shown in table 3. It reduces the size of the RBF training data to 4x4500 and 9x4500.

Table 3: Faults considered for dynamic FDI

S. No.	Fault
1	No Fault (NF)
2	10% air leakage in intake manifold
3	20% air leakage in intake manifold
4	EGR valve stuck in 50% closed position
5	EGR valve stuck in 100% closed position
6	Temperature sensor 20% over reading
7	Temperature sensor 20% under reading
8	Pressure sensor 20% over reading
9	Pressure sensor 20% under reading

MATLAB could easily handle this data size and an RBF is trained with a spread constant of 0.5 and error threshold of 300, which results in a network of 65 neurons. Subsequently the trained RBF network is tested for different sets of seen and unseen data as shown in table 4.

Table 4: Testing data types

Initial throttle angle	Final throttle angle	Remarks
22	22	Seen Data
26	26	Seen Data
38	30	Seen Data
24	30	Partially seen data
28	32	Unseen data
36	28	Unseen data

Test results show that the trained network is very well able to interpolate the unseen data and classifies all the faults with 0% misclassification. The result for the faults occurring when initial condition of throttle angle is 28^o and final condition is 32^o, are shown in fig 10.

Practically there can be infinite number of possibilities for initial and final conditions of the engine. The network is trained for the above five equally distributed typical cases. The network interpolates the remaining in-between situations and results in proper fault classification.

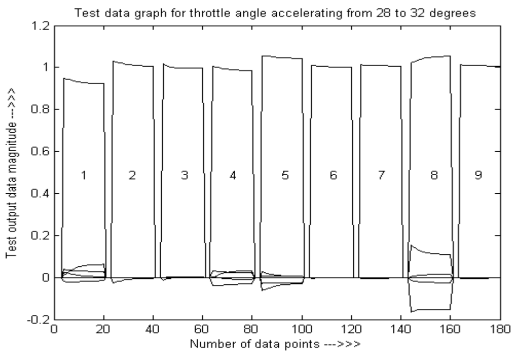

Fig 10: RBF test result for eight faults case 3

7. CONCLUSIONS

A different neural network approach for fault diagnosis and isolation in automotive engines using dynamic data is investigated in this paper. The technique is demonstrated for two different conditions with results of 0% misclassification. It is further used for a more general case as discussed in section 6.3. In this case, due to computer memory limitation, five different neural networks are trained and tested for 0% misclassification for all the 17 fault conditions. But these five independent neural networks were not able to interpolate the unseen data to a satisfactory level of acceptance. Further, a single neural network is trained for nine fault conditions in order to reduce the size of the training data. The trained RBF neural network is found to interpolate the in-between values of the data very well and is tested for unseen data sets, which resulted in 0% misclassification. It is therefore recommended to have a single large neural network instead of five smaller separate networks. The reason is that a single neural network interpolates the in-between values, which it has not seen before, in a much better way than the five separate neural networks.

Further work of developing algorithms for MATLAB to handle large data matrices in network training is required to extend the investigations. Testing and development for multiple faults, unseen fault intensity classification, detection of unknown faults, performance with real data and engine parameter variations are also required.

8. NOMENCLATURE

The following nomenclature is used in this paper:

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

REFERENCES

Chen S., Grant P. M., and Cowan C. F. N., (1991), Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Trans. Neural Networks*, **2**, 302-309.

Cook J.A., Powell B.K., (1993), Modelling of an internal combustion engine for control analysis, *IEEE Control Systems*, **13(3)**, 62-68.

Gomm J. B., Weerasinghe M. and Williams D., (2000), Diagnosis of process faults with neural networks and principal component analysis. *Proc I MechE Part E*, **214**, 131-143.

Hendricks E., (2001), Isothermal vs. adiabatic mean value SI engine models. *Proc 3^rd^ IFAC workshop,* 28-30 Mar. 2001, 363-8.

Hendricks E., Engler D. and Fam M., (2000), A generic mean value engine model for spark ignition engines. Institute of Automation, Denmark Technical University. www.iau.dtu.dk/~eh/.

Hendricks E., Chevalier A., Jensen M., Sorenson S. C., Asik J. and Trumpy D., (1996), Modelling of the intake manifold filling dynamics, *SAE* Technical Paper No. 960037.

Hendricks E., Vesterholm T., Kaidantzis P., Rasmussen P. B. and Jensen M., (1993), Nonlinear transient fuel film compensation, *SAE* Technical Paper No. 930767.

Laukonen E. G., Passino, K.M., Krishnaswami V., Luh G. C. and Rizzoni G. (1995), Fault detection and isolation for an experimental internal combustion engine via fuzzy identification. *IEEE Trans on Control Systems Technology*, **3(3)**, 347-355.

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

Sangha M. S., Gomm J. B., Yu D. L., Page G. F. (2004), An investigation of neural networks in fault diagnosis of an automotive engine air path, Proc. 2nd ASIM Workshop Wismar– Modelling, control and diagnostics of combustion engine processes, Germany, September 16-17, 2004, 23-32, ISBN 3-901608-27.

Sorsa, T. and Koivo, H. N., (1993), Applications of artificial neural networks in process fault diagnosis. *Automatica*, **29(4)**, 843-849.

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

Yong-Wha Kim, (1998), Automotive engine diagnosis and control via nonlinear estimation, *IEEE Control Systems*, 84-99.