Investigation of the Different Type Models for Engine Test Data Modelling

M.H. Wu
School of Technology, University of Derby,
Marketon St., Derby, DE22 3AW, UK

Abstract
The engine test bed has been used widely in automotive industry to test the protetypey developed Engine in order to understand the engine operation process and find the optimum parameters for the engine operation. As a result, thousand and thousands data will be obtained from Engine Test Bed.for this purpose. It is a time consuming task to get the data and it is hard to analyse those data directly from the huge number of the data. The best way to reduce the tested data from the test bed is to create a mathematics model from the limited engine tested data.
The paper will list three different models used for the Engine Test Data modelling, Polynomial model, Combined Exponational model and Neural Network model. The principle and equations of each model has been introduced in the paper. The experiment results of each model have been listed as well. The results from last two developed models show that the models provide the achievement which can meet the customers requirement.
Keywords Engine test data, Engine performance, Engine data modelling, Neural network

1 Introduction

The engine test bed has been used widely in automotive industry to test the protetypey developed Engine in order to understand the engine operation process and find the optimum parameters for the engine operation. As a result, thousand and thousands data will be obtained from Engine Test Bed.for this purpose. It is a time consuming task to get the data and it is hard to analyse those data directly from the huge number of the data.

The best way to reduce the tested data from the test bed is to create a mathematics model from the limited engine tested data. The model will be used to analyse the Engine operation and to obtain the optimum operation data. Figure 1 shows the structure of the Engine Test Bed (ETB). The newly developed or existed engines should be tested on it. Fig.1 shows the format of the tested data from ETB which are used to create the new model. The controled input variables and the responding out put are shows in Fig.2.

It includes: Ignition time, Ex valve timing, Inlet valve timing , BMEP and Speed as the input variable and the BSFC as the output.

There are three models developed for the Engine Data Modelling. It is Poly-

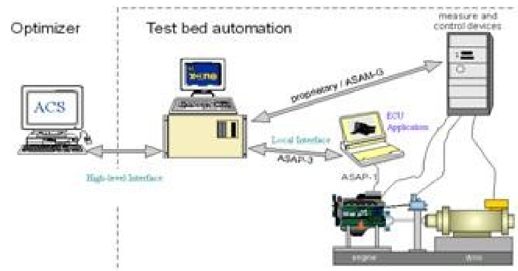


Fig.1 Structure of Engine Test Bed

nomial model, Combined Exponational model and Neural Network model.

NG4 Bg2 modelling

input 1 Speed	input 2 BMEP	input 3 VanosIn	input 4 VanosEx	input 5 Igtiming	Output 1 g/kwhr BSFC
1984	6.7	105	95	18	263.0
2234	5.7	92	83	25	263.7
2356	5.8	118	83	23	264.2
1984	6.5	105	95	16	264.8
2235	6.0	119	107	20	265.0
2234	5.8	92	107	23	265.5
2235	5.9	119	107	18	266.1
2357	5.9	118	106	20	266.7
2357	5.9	92	108	24	266.7
2736	6.6	105	95	20	268.7
2235	5.8	92	107	20	269.0

Fig.2 The form of the tested data from the ETB

2 Employing the Polynomial Model for Engine Test Data Modelling

2.1 The Polynomial Approximation Model

(1)shows the aplication of the polynomial approximation model in engine test data modelling. It called "Quadratical Model"(QM) which is denoted by Voigt, Lechner and Hochschwarzer[1-3].

The QM is:

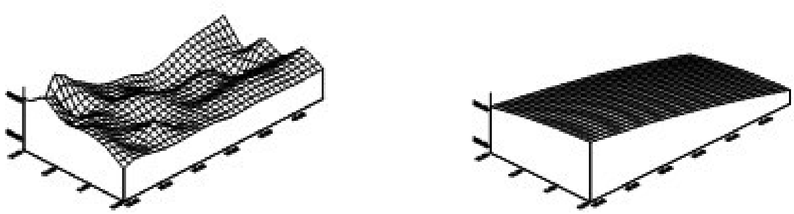
$$y = x^T Q x + a^T x + b \tag{1}$$

here: y is an objective vector,
x is an input variable vector,
Q is co-efficient matrix,
a& b are constant vectors,

The co-efficient of matrix Q and the vectors a and b are calculated by least-squares-method to minimize the deviation of measured values from the model values.

2.2 Analysis of the Application of the Polynomial Model

This model has the advantage of simplicity, but is imperfect for real application. Fig.3 shows the example which indicates the model's imperfect.



(a) draw by the Original engine tested data

(b) draw by the data calculated by QM

Fig.3 3D surface draws by the different data

Fig.3a shows the relationship between the BSFC and two input variables by the experiment data directly. Fig.3 b shows the same relationship using the model QM. It is obviously that Fig.3(a) is totally different to Fig.3(b). In Fig.3a it shows that the output will be varied as a irrgular wave when two inputs are varied. The Fig.3b only provide a smooth raised convex area. In one word, the QM model can not trace the features in Fig.3a. It is clearly that QM is not a suitable model to be used for engine parameters modelling and optimasation. The data from QM provides the large errors and losts nearly most of data features. Another disadvantage of using quadratical equations is that the model only can be used to the case which has one convex (or concave) data set.

3 Employing the Combined Exponational Model for Engine Test Data Modelling

3.1 The Combined Exponational Model

(2) shows the aplication of the combined exponational model in engine test data modelling[4].

$$f(x_1, x_2, \dots x_n) = a + \sum_{i=1}^{n} b_i x_i + \sum_{j=1}^{p} (h_j E(x_1) E(x_2) \dots E(x_n)) \tag{2}$$

here: $E(x) = exp(-8(x-m)^2/w^2)$;

h : the heigth of the peak, it could be "+" or "-";

m : the position of the peak along the axis;
w : the width of the peak;
n = 5, (the model dealings the variables;
$f(x_1, x_2, ...x_n)$: break specific fuel consumption (BSFC);
x_1 : ignition time variable;
x_2 : inlet valve opening degree variable;
x_3 : exhaust valve opening degree variable;
x_4 : speed;
x_5 : toque (BMEP);
p : the number of peaks in nD space; it assumes p = 10;
a : a constant;
b_i : a slope coefficient respect to input xi;

Fig.3a indicts that the output BSFC could be formed by constant element, linear element and multi- convex or concave peaks. The convex or concave peaks on the 3D surface can be expressed by the function of exponatioal.
As a result, the (2) can be explained as follow:

BSFC = Constant + Linear function + Exponational function;

3.2 Set Up the Coefficients of the Combinated Exponational Model

If the equation 2 has been set by 20 peaks (j=20) and 5 input variables (i=5), the total number of the coefficients of the equation, b_i, h_j, $m_{i,j}$ and $w_{i,j}$, should be as following Table 1.

There is no suitable mathematics methods to set up the exactly values to 225

Table 1 The number of the parameters of the combined exponational model with 5 inputs, 20 peaks

Coefficient	b_i	h_j	m_{ij}	w_{ij}	Total
The numbers	5	20	100	100	225

coefficients through the existed tested data. As a result, Genetics Algorithm are used to define the appropriated values for each coefficient by the existed tested data. A VB based program is developed for this purpose. The software will be used to obtain the appropriate values of the coefficients in (2) by using the follow operations in GA:

– random selection of the chromosome within certain range,
– reproduction
– crossover
– evaluation
– mutation

In order to running the GA, the follow assumptions and definitions are re-

quired:
Gene: the genes are used to form the chromosome in GA application. As a result, the number of the Gene should be equal to the total numbers of the coefficients in the (2). For above example in table 1, the numbers of gene should be 225.
Population: the population is the numbers of the chromosome. It can be set from 1 to 100 in the software. The rule of setting up is that the more coefficients is , the larger population should be. In the case of table 1, the population is settled equal to 100.
Number of keep best members of the population: this is very important selection. If it is too small, too many poor chromosomes will be remained in the operator. On the other hand , it is easily tripped into the local best point. For the case in table 1, a range 10-20 is suggested.
Crossover method: crossover method can be selected from alternative methods, single point cross over, two points crossover, etc.
Type of mutation: Random mutation hill climb and directional hill climb methods are available in the software. The first one is normally used for the fitness function is highly discontinuous. The 2^{nd} one is for the continuous fitness function. Our fitness function for GA is continuous so that the directional hill climb method is adopted.
Mutation probability of population: This percentage is employed to avoid trapped in local best. For engine test data modelling, the percentage is 10%.
Mutation probability of genes: If the percentage is too high, many "wild" mutations which have very poor fitness will be involved. If it is too low, some necessary genes will not be involved in crossover operator to improve the solution. It is normally 8-10%.
Fitness function in evaluation operation: Fitness function is very important for chromosome parent selection in GA operation
Equation 3 is the fitness function used in the GA evaluation.

$$\min f(x_1, x_2, ..., x_n) = \sum \left(a + \sum_{i=1}^{n} b_i x_i + \sum_{j=1}^{p} h_j E(x_1) \cdot E(x_2) \cdots E(x_n) - y_0 \right)^2 \tag{3}$$

Defined the ranges of variation of each variable: The ranges of variation of each variable in the engine test bed is settled as table 2, which will constraint the variables of x_1, x_2,...,x_n; in (2) during the GA operation.

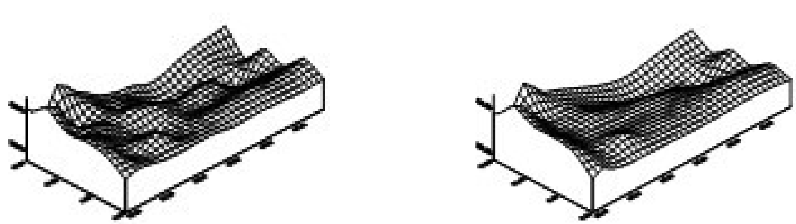
3.3 Output Results From the Combined Exponationa Model

As mentioned before, the parameters of the (2) are obtained from the existed test data from ETB by GA system. As soon as the parametes is defined, the equation

Table 2 Experimental data ranges

Variables		Range
X1	Input: speed	1237 to 4253
X2	Input: BMEP	0.9 to 6.8
X3	Input: VanosIn	80 to 132
X4	Input: VanosEX	70 to 135
X5	Input: Igtiming	4.5 to 51
Y	Output: BFSC	Average: 346

2 can be used to calculate the engine output BSFC. Fig. 4 shows the 3D surface drawings by the ordinary test data and the data calculated from (4) It is obviously that the major convex or concave peaks in (c) are repeated in (d). It means that the new model can generat the major features of the black box system (Engine Perfomance) and it is much more better than using the Quadratical Model in Engine Test Data modelling.



(a) draw by the Original tested data

(b) draw by the calculated data from new model

Fig.4 comparing the Original tested data and the calculated data using the new model

4 Employing the Neural Network Model for Engine Test Data Modelling[5-6]

4.1 Develop the Radial Basis Function (RBF) Networks for Engine Calibration Modelling Task[5]

In 2003, Dr. Lin has published a paper. In the paper the neural networks used on the Engine Test data modelling was mentioned. On the application, Radial Basis Function (RBF) Networks is used for Engine Calibration modelling task. RBF networks are universal approximates, that is, given a network with enough hidden layer neurones, they can approximate any continuous function with any arbitrary accuracy [7-8].

An RBF is one whose output is symmetric around an associated centre μ_j. It is generally described as:

$$\phi_j(r) = \phi_j(r)(\|x - \mu_j\|);\ x \in R^n; r \geq 0 \tag{4}$$

where $\phi_j(r)$ is a continue function on$(0, \infty)$and $\|\cdot\|$ denotes the Euclidean norm. Linear combination of RBFs represents a wide classes of functions:

$$y_j(x) = \sum_{j=1}^{M} \omega_{ij}\phi_j(\|x - \mu_j\|) + \omega_{i0} \tag{5}$$

The bias w_{i0} compensates for the difference between the average value over the

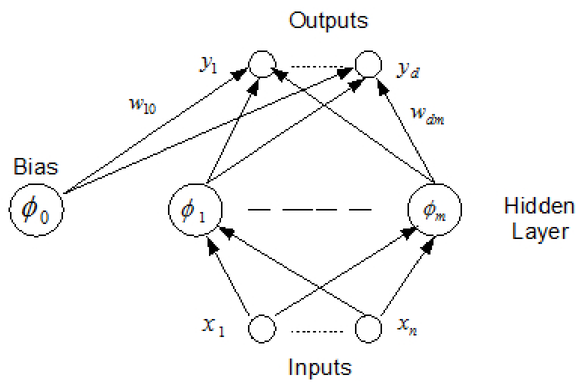


Fig.5 RBF network structure

data set of the basis function activation and the corresponding average value of the targets. The RBF network is depicted as shown in Fig.5. The bias w_{i0} can be incorporated into the summation by introducing an extra basis function ϕ_0 and setting its activation to unity. Thus the (2) can be written as

$$y_j(x) = \sum_{j=1}^{M} \omega_{ij}\phi_j(\|x - \mu_j\|) \tag{6}$$

Follows are the typical functions of the RBF:

- The Gaussian function:

$$\phi(r) = \exp(-r^2/2) \tag{7}$$

- The thin plate spline function:

$$\phi(r) = r^2 \times \log r \tag{8}$$

- The multiquadric function:

$$\phi(r) = (r^2 + 1)^{1/2} \tag{9}$$

- The inverse multiquadric function:

$$\phi(r) = \frac{1}{(r^2 + 1)^{1/2}} \tag{10}$$

- The pseudo cubic spline function

$$\phi(r) = r^3 \tag{11}$$

- The logarithmic function:

$$\phi(r) = \log(r^2 + 1) \tag{12}$$

where r is a non-negative number and is the scaled distance from the input vector x to the RBF centre , which is defined as :

$$r_j = \sqrt{\sum_{l=1}^{n} \frac{(x_l - \mu_{lj})^2}{\sigma_j^2}} \tag{13}$$

In (13), n is the dimension of input vector and σ_j is scale factor or width of RBF. In Engine Calibration modelling task, all of six RBFs described above are applied.

4.2 Develop Neural Network Modelling Tool for Engine Calibration Modelling Task[6]

The Fig.6 shows the structure of the Neural Network Modeling Tool used on the Engine Test Bed. When the tested data is ready, they will be sending into the NN Models Tool Box. In the NN tool box, there are three different types and totally ten NN structures. In it, the total tested data will be divided into three group. The first group data will take 50 % out of the total tested data. The second and third groups will take 25% out of the total data respectively. The first group data will be used to train the neural network and second group data will used to validate the NN model. Final group data will be used to test the NN model.

After NN Tool Box, the best fitted NN structural model is selected and sends to optimization section to defined the optimization data for Engine Management Unit.

In the optimization section, there are two options for optimization purposes according to the different requirement. SOGA is the model for single objective optimization and MOGA is one for multi objectives optimization.

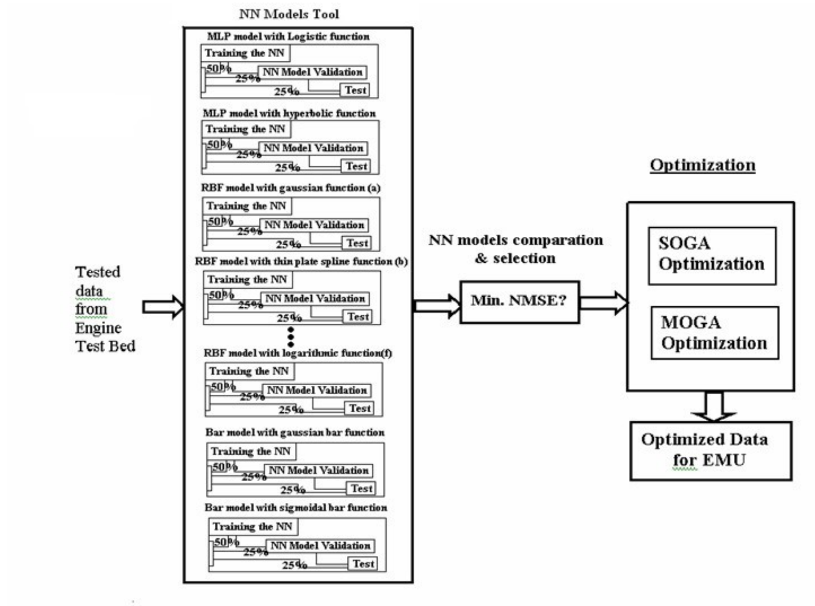


Fig.6 The structure of the optimisation system of the engine performance

As shown in Figure 6 the NN model tool is employed to select a best suitable NN model out of ten NN models for the subsequence operation of optimization.

Fig.7 shows the common structure of the neural network in NN model Tool and it is under the following assumptions:

- it is a three layers NN model.
- the numbers of the hidden layer note is defined by 2n + 1: here n is the number of inputs.

The NN Model Tool includes three different type of neural networks, Multi-layer perceptron (MLP); radial basis functions (RBF) and bar function (BAR).

a. MLP. The output of the MLP is denoted by

$$\begin{aligned} y_k(x) &= \sum_{j=0}^{m} f(\sum_{i=0}^{n} x_i\omega_{ji})\omega_{kj} \\ &= \sum_{j=1}^{m} f(\sum_{i=1}^{n} x_i\omega_{ji} + \omega_{j0})\omega_{kj} + \omega_{k0}, \quad k = 1, \ldots, l \end{aligned}$$

where ω_{ji} and ω_{kj} are the input-hidden weight and hidden-output weight, respectively. f is the activation function which has two types:

(a) logistic function

$$f(\alpha) = \frac{1}{1 + e^{-\alpha}} \tag{15}$$

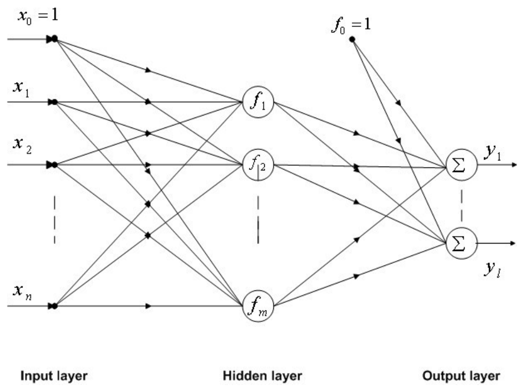


Fig.7 Neural network structure

(b) hyperbolic tangent function or tanh

$$f(\alpha)=\frac{e^{\alpha}-e^{-\alpha}}{e^{\alpha}+e^{-\alpha}} \tag{16}$$

b. RBF. The output of the RBF is calculated using the following form in (5) and (6):

$$\begin{aligned} y_k(x) &= \sum_{j=1}^{m}\phi(\|x-\mu_j\|)\omega_{kj}+\omega_{i0} \\ &= \sum_{j=0}^{m}\phi(\|x-\mu_j\|)\omega_{kj}, \quad k=1,\ldots,l \end{aligned}$$

where ω_{kj} is the hidden-output weight and μ_j is the centre of j-th hidden unit. ϕ is the kernel function. To defining a distance from the input vector $\mathbf{x}$ to the RBF centre μ_j scaled by the scale factor or width σ_j , equation 13 is used.
Within the RBF networks, six kernel functions are widely used and defined by equations from (7) to (12).
c. BAR. The BAR network has the same structure as the RBF network, but the kernel function is different. Two kernel function for the BAR network:
(a) Gaussian bar function

$$\phi(x)=\sum_{i=1}^{n}\exp\Big[-\frac{(x_i-\mu_{ji})^2}{2\sigma_{ji}^2}\Big] \tag{17}$$

(b) sigmoidal bar function

$$\phi(x)=\sum_{i=1}^{n}1/\Big\{1+\exp\big[-\frac{(x_i-\mu_{ji})^2}{2\sigma_{ji}^2}\big]\Big\} \tag{18}$$

Table 3 lists the summery of the neural network structure used in the NN Model Tool.

Table 3 Nomenclature of neural network structures

NN Model	Activation Function	Abbreviation
RBF	Gaussian function	Rbf-Gaussian
	thin plate spline function	Rbf-TPS
	logarithmic function	Rbf-Logarithmic
	multi-quadric function	Rbf-Quadric
	inverse multi-quadric function	Rbf-InverseQuadric
	pseudo cubic spline function	Rbf-Cubic
BAR	Gaussian function	Bar-Gbar
	Sigmoidal function	Bar-Sbar
MLP	logistic function	Mlp-Logsig
	tangent function	Mlp-Tansig

4.3 Experimental Results

4.3.1 Experiment Results From the RBF NN Model

A set of engine test data with three inputs variables and one output variable was employed to evaluate the performance of the proposed approach. The input variables are "break mean efficient pressure"(BEMP), "inlet valve opening degree"(VanosIn), and "exhaust valve opening degree" (VanosEx). The output is "break specific fuel consumption"(BSFC). The whole set of data are from 293 test points. The 196 data points out of 293 are used for training the GA-RBF network and another 97 data points are used for test the GA-RBF network.

In the experiment, it run under the following conditions and definitions:

- the number of RBFs is fixed at 20
- For comparison purposes, the six different RBF networks are executed 10 times
- the normalised mean squares error (NMSE) is used as a performance measure for the different RBFs:

$$NMSE = \frac{\left(\frac{1}{K}\sum_{i=1}^{K}(y_i - t_i)^2\right)^{1/2}}{\left(\frac{1}{K}\sum_{i=1}^{K}(t_i - \bar{t})^2\right)^{1/2}} \tag{19}$$

where y_i and t_i are respectively the model output and the target value, and $\bar{t}$ is the mean value of the target values on the training data set or test data set. This expression has the value 0 for a perfect match between model and target, and the value 1 if the model just outputs the target mean $\bar{t}$.

Fig.8 shows the evolutionary process of different RBF.

Fig.8(a) is the average evolutionary process of 10 runs over the training data for six RBFs and Fig.8(b) is the best process of 10 runs. The results review that the Gaussian function provides the better results compare with other RBFs in terms of convergence speed and modelling accuracy. It is also shown that the local basis function has a faster convergence speed than non-local basis function.

The Table 4 summaries the final average NMSE of 10 runs for different RBFs. The result indicates that the Gaussian function has the smallest NMSE while the pesudo cubic spline has the largest NMSE.

Table 4 Final average NMSE over 10 runs for training data

	Average NMSE
Gaussian	0.1217
Thin Plate Spline	0.1397
Multiquadric	0.1304
Inverse multiquadric	0.1368
Pseudo Cubic Spline	0.1508
Logarithmic	0.1274

Table 5 compares the final best NMSE of 10 runs for both training data and test data. It indicates clearly that the Gaussian function performs better than other RBFs in both training data and test data.

The figures from Fig.5 to Fig.10 plot the modelling performance of both training data and test data using the best final parameters of RBF networks by RCGA run. These figures have shown that the RBF networks training by RCGA are successfully applied in modelling of engine test data.

Table 5 Final best NMSE over training data and test data

	Training data	Test data
Gaussian	0.1134	0.1119
Thin Plate Spline	0.1231	0.1297
Multiquadratic	0.1268	0.1245
Inverse Multiquadratic	0.1304	0.1281
Pseudo Cubic Spline	0.1286	0.1419
Logarithmic	0.1235	0.1240

Fig.9 shows the results.of RBF network model for real engine test data analysis and modelling. The model provides a close fit to the ordinary tested data successfully.

4.3.2 Experiment Results From the NN-Tool

In this section, an experiment example for engine data modelling was carried out. The input data and output data are listed in Fig.10 and Fig.11.

Three NNs, MLP, RBF and BAR, with different activation functions, abbrevi-

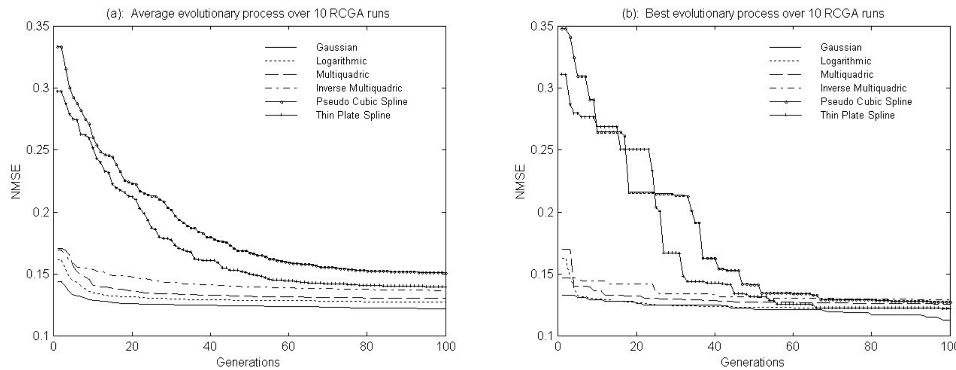
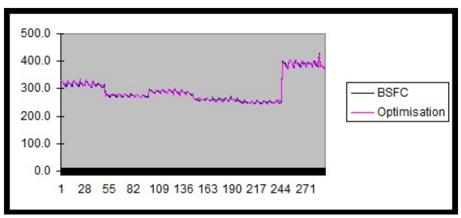


Fig.8 Evolutionary process of RBF networks over 10 runs



Error max	10.4	4.0%
Error min	-10.1	-3.1%
Fitness	28403	

Fig.9 Final results

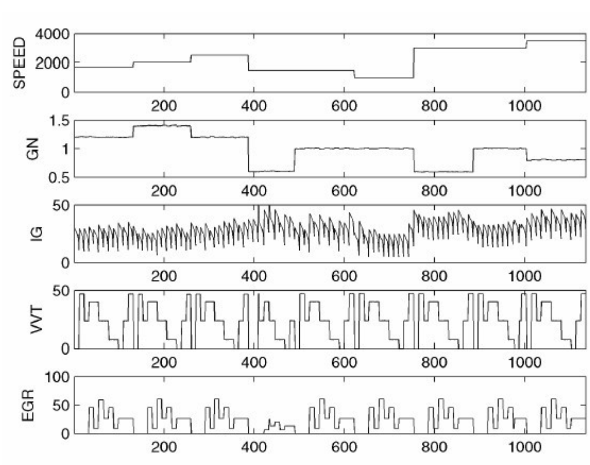


Fig.10 Input data set: SPEED (rpm), GN(%), IG(degree), VVT(degree) and EGR(%)

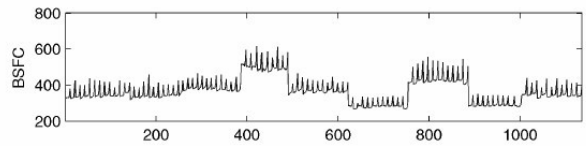


Fig.11 Output data set: BSFC (g/kWh)

ated in Table 3, were used throughout the example. The data was taken from engine test bed of the Lander Rover Group, plc.

In order to assess the goodness of modelling, the normalized mean squared error (NMSE) over each output variable is used:

$$NMSE = \frac{\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2\right)^{1/2}}{\left(\frac{1}{N}\sum_{i=1}^{N}(t_i - \bar{t})^2\right)^{1/2}} \tag{20}$$

where N is the number of total data points. y_i and t_i are respectively the model output and the target value, and $\bar{t}$ is the average value of the targets on the data set, defined as:

$$\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{21}$$

NMSE has the value 0 for a perfect match between model and target, and

Table 6 Mean, standard (std. dev), minimum and maximum MSE over 10 runs for each model

	Mean	Std.dev.	Minimum	Maximum
	Training			
Rbf-Gaussian	278.83	34.82	233.46	330.88
Rbf-TPS	283.34	32.02	237.32	308.40
RbF-Logarithmic	277.39	19.58	241.20	304.25
Rbf-Quadratic	292.43	7.45	288.74	313.35
Rbf-InverseQuadratic	290.63	59.92	229.97	386.28
Rbf-Cubic spline	292.55	3.56	289.18	295.93
Bar-Gbar	66.75	11.82	56.47	84.41
Bar-Sbar	68.65	14.98	54.77	88.86
Mlp-Logsig	269.44	67.78	158.40	339.92
Mlp-Tansig	290.94	19.86	237.06	303.64
	Validation			
Rbf-Gaussian	289.49	27.02	230.97	327.41
Rbf-TPS	301.45	17.05	282.35	323.80
Rbf-Logarithmic	279.53	18.92	255.86	308.97
Rbf-Quadratic	300.01	11.51	287.04	316.40
Rbf-InverseQuadratic	301.84	35.50	263.44	353.18
Rbf-Cubic spline	300.42	4.91	295.76	305.08
Bar-Gbar	81.79	12.43	71.06	100.77
Bar-Sbar	85.66	17.06	70.20	106.74
Mlp-Logsig	281.66	74.78	145.68	341.73
Mlp-Tansig	278.11	5.86	261.45	280.15

	Mean	Std.dev.	Minimum	Maximum
	Test			
Rbf-Gaussian	335.57	34.57	274.02	379.63
Rbf-TPS	345.90	24.43	310.51	361.82
Rbf-Logarithmic	316.09	19.19	292.00	346.66
Rbf-Quadratic	337.71	12.71	326.24	351.81
Rbf-InverseQuadratic	348.45	44.95	308.03	424.94
Rbf-Cubic	323.60	8.68	315.36	331.83
Bar-Gbar	85.16	12.59	74.72	106.83
Bar-Sbar	92.93	19.93	74.35	118.90
Mlp-Logsig	304.54	75.97	162.30	364.69
Mlp-Tansig	341.02	17.74	290.52	346.77

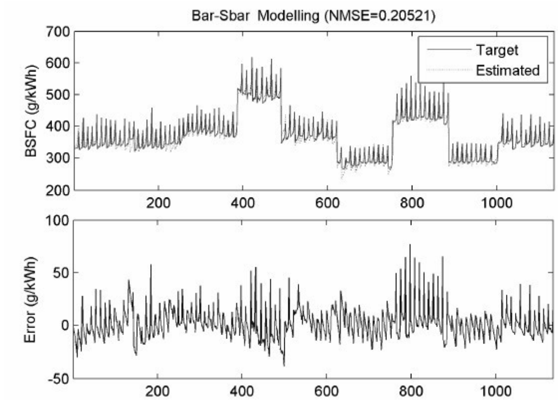


Fig.12 Modelling performance of BSFC by sigmoidal BAR model

the value 1 if the model just outputs the target mean$\bar{t}$ .Table 6 shows the result of the example. It is clearly that the model structure which has the minimum MSE was the BAR model with sigmoidal bar function. The modelling result by Bar-Sbar is plotted in Fig.12.

5 Conclusion

In this paper, the models used by the Engine Test data modelling system is introduced. The details of RBF network and the Neural Network Tool have been introduced and the experiment results show that the last two models are running successfully to catch the relationship between the input data and output data within the Engine tested Data.

References

[1] Voigt K.U. (1993), Model based on line optimization for modern engine management system, SAE Brazil.

[2] Lechner H. (1996), Automated system for optimized calibration of engine management systems, Rover Group internal publication circa.

[3] Hochschwarzer H. (1992), Fully automatic optimization of engine calibration, *IMecE proceeding,* pp.89-100.

[4] W C Lin & M H Wu. (2003), Optimization using genetic algorithm in engine calibration, ICMR2003, Glasgow, UK.

[5] C Lin, M H Wu, S Y Duan. (2003), Engine test data modelling by evolutionary radial basis function networks, IMechE publication, *Journal of Automobile Eng, Part D,* Vol.217, pp.489-497.

[6] M H Wu, Wanchang Lin, S Y Duan. (2008), Develop a nn modelling tool for engine performance tested data modelling, ICMR2008, Brunel University, U.K.

[7] Park, J. and Sandberg, I. W. (1991), Universal approximation using radial basis function networks, *Neural Computation,* 3:pp.246-257.

[8] Poggio T. and Girosi F. (1990), Networks for approximation and learning, *Proceedings of IEEE,* Vol.78, pp.1481-1497

Corresponding author
Author can be contacted at: m.h.wu@derby.ac.uk