DIESEL ENGINE PERFORMANCE MODELLING USING NEURAL NETWORKS

by

MARK STEVE RAWLINS

DISSERTATION SUBMITTED IN COMPLIANCE WITH THE REQUIREMENTS FOR THE DOCTOR'S DEGREE IN TECHNOLOGY: ENGINEERING: MECHANICAL

FACULTY OF ENGINEERING, SCIENCE AND THE BUILT ENVIRONMENT
DEPARTMENT OF MECHANICAL ENGINEERING

DURBAN, KWAZULU NATAL, SOUTH AFRICA
JUNE, 2005

SUBMISSION APPROVED FOR EXAMINATION BY SUPERVISOR

_____ _____

Prof. P Tabakov, PhD (UN) Date

(Associate Professor, Department of Mechanical Engineering. – DIT)

DURBAN INSTITUTE OF TECHNOLOGY
FACULTY OF ENGINEERING, SCIENCE AND THE BUILT ENVIRONMENT
DEPARTMENT OF MECHANICAL ENGINEERING

I hereby declare that this dissertation is my own unaided work except where due acknowledgement is made to others. This dissertation is being submitted to the Durban Institute of Technology for the Degree of Doctor of Technology, and has not been submitted previously for any other degree or examination.

_____ _____

Mark Steve Rawlins (18600346) Date

Abstract

The aim of this study is to develop, using neural networks, a model to aid the performance monitoring of operational diesel engines in industrial settings. Feed-forward and modular neural network-based models are created for the prediction of the specific fuel consumption on any normally aspirated direct injection four-stroke diesel engine. The predictive capability of each model is compared to that of a published quadratic method.

Since engine performance maps are difficult and time consuming to develop, there is a general scarcity of these maps, thereby limiting the effectiveness of any engine monitoring program that aims to manage the fuel consumption of an operational engine. Current methods applied for engine consumption prediction are either too complex or fail to account for specific engine characteristics that could make engine fuel consumption monitoring simple and general in application. This study addresses these issues by providing a neural network-based predictive model that requires two measured operational parameters: the engine speed and torque, and five known engine parameters. The five parameters are: rated power, rated and minimum specific fuel consumption bore and stroke.

The neural networks are trained using the performance maps of eight commercially available diesel engines, with one entire map being held out of sample for assessment of model generalisation performance and application validation. The model inputs are defined using the domain expertise approach to neural network input specification. This approach requires a thorough review of the operational and design parameters affecting engine fuel consumption performance and the development of specific parameters that both scale and normalize engine performance for comparative purposes. Network architecture and learning rate parameters are optimized using a genetic algorithm-based global search method together with a locally adaptive learning algorithm for weight optimization. Network training errors are statistically verified and the neural network test responses are validation tested using both white and black box validation principles. The validation tests are constructed to enable assessment of the confidence that can be associated with the model for its intended purpose.

Comparison of the modular network with the feed-forward network indicates that they learn the underlying function differently, with the modular network displaying improved generalisation on the test data set. Both networks demonstrate improved predictive performance over the published quadratic method. The modular network is the only model accepted as verified and validated for application implementation.

The significance of this work is that fuel consumption monitoring can be effectively applied to operational diesel engines using a neural network-based model, the consequence of which is improved long term energy efficiency. Further, a methodology is demonstrated for the development and validation testing of modular neural networks for diesel engine performance prediction.

Acknowledgements

My gratitude and thanks to the following people and institutions:

Professor P Tabakov for his valuable support as my supervisor.

Professor K Duffy for his valuable contribution, both in the modelling methodologies and technical proofing.

Annemarie Rawlins and the library staff of the Alan Pittendrigh Library on the Steve Biko Campus of the Durban Institute of Technology.

The Department of Mechanical Engineering for their assistance over the years.

The directors and staff of Energy and Combustion Services (Pty) Ltd who assisted with many computing resources, information, and time for this work.

Dedication

To Annemarie, Paul and Maia
for your support, patience and understanding

Table of Contents

CHAPTER 4

CHAPTER 5

CHAPTER 7

CHAPTER 8

List of Tables

List of Figures and Illustrations

List of Symbols, Nomenclature and Abbreviations

Symbol	Description	Dimensional Units
A_p	Engine piston area	m^2
C	Number of elements in output layer and contour number	
$\mathrm{CI}_{(1-\alpha)}, \mathrm{CI}_{\mathrm{M}}$	Confidence Interval	
F_f	Frictional force	N
F_i	Input fan-in	
M_c	Integrated fuel mass (Total fuel consumed)	kg
G_n	Genotype	
H_a	Alternative hypothesis	
H_0	Null hypothesis	
I	Number of neural network input parameters	
$IW_{j,i}$	Input weights	
J	Number of hidden processing units in the neural network	
K	Number of output units in the neural network	
$\mathrm{LCL}_{\mathrm{R}}, \mathrm{UCL}_{\mathrm{R}}$	Lower and Upper Confidence Levels	
$LW_{j,i}$	Layer weights from the l^{th} neuron to the j^{th} neuron	
M	Set of non-linear basis functions	
M_c	Integrated fuel mass flow	kg
M_{ca}	Actual fuel consumption	kg
M_{ct}	Target fuel consumption	kg
N	Engine rotational (angular) speed	$rev.s^{-1}$
N	Total training epochs	
N	Total input-output vectors	
N_{train}	Number of network training exemplars	
P_i	Indicated mean effective pressure (imep)	kPa or $N.m^{-2}$ $x10^3$
P_f	Frictional mean effective pressure (fmep)	kPa or $N.m^{-2}$ $x10^3$
P_{fm}	Mechanical friction mean effective pressure	kPa or $N.m^{-2}$ $x10^3$

Symbol	Description	Dimensional Units
P_b	Brake mean effective pressure (mep)	kPa or $N.m^{-2}$ $x10^3$
$P_{b_{\text{rated}}}$	Rated brake mean effective pressure	kPa or $N.m^{-2}$ $x10^3$
$Q_{1,p}$	Heat addition at constant pressure (ideal cycle)	J/cycle
$Q_{1,V}$	Heat addition at constant volume (ideal cycle)	J/cycle
Q_2	Heat to exhaust (ideal cycle)	J/cycle
Q_c	Specific fuel consumption	g/kW.h
$Q_{c\ \text{rated}}$	Rated specific fuel consumption	g/kW.h
$Q_{c_{\text{nom rated}}}$	Nominal rated specific fuel consumption	g/kW.h
$Q_{c\ \text{min}}$	Performance Map Minimum specific fuel consumption	g/kW.h
$Q_{c\ \text{std}}$	Standard rated specific fuel consumption	g/kW.h
R_{AV}	Ratio of piston area to displaced volume	m^{-1}
S_n	Total network weight parameters	
T_{ca}	Fuel temperature at the flow meter	℃
U_p	Piston speed	$m.s^{-1}$
$\bar{U}_p$	Mean piston speed	$m.s^{-1}$
$\bar{U}_{p_{rated}}$	Rated mean piston speed	$m.s^{-1}$
V	Volume	m^3
V_d	Volume displaced per cycle	dm^3
$W_{c,i}$	Indicated work done per cycle	J
W_{ki}, W_{kj}, W_{ji}	Processing element weight parameters	
W_{j0}, W_{k0}	Bias parameters	
$\dot{W}_{c,i}$	Indicated power per cycle	kW or $J.s^{-1}$ $x10^3$
$\dot{W}_i$	Indicated power	kW or $J.s^{-1}$ $x10^3$
$\dot{W}_f$	Frictional power	kW or $J.s^{-1}$ $x10^3$
$\dot{W}_b$	Brake power	kW or $J.s^{-1}$ $x10^3$
$\dot{W}_A$	Specific power	kW. m^{-2}
$\dot{W}_{A_{\text{rated}}}$	Rated specific power	kW. m^{-2}
$\dot{W}_{A_{\text{min}}}$	Minimum specific power	kW. m^{-2}

Symbol	Description	Dimensional Units
X_{N}	Normalised mean piston speed (quadratic method)	
Y_{N}	Normalised mean effective pressure (quadratic method)	
$\overline{Y}_{\mathrm{M}}$	Model response (output) mean	
$\overline{Y}_{\mathrm{R}}$	Target response mean (real world or observed)	
Z_{N}	Normalised specific fuel consumption (quadratic method)	
$\mathbf{x}$	Input vector	
$\mathbf{x}_{\mathrm{R}}, \mathbf{x}_{\mathrm{M}}$	Real world (observed) and model input vectors	
$\mathbf{y}_{\mathrm{R}}, \mathbf{y}_{\mathrm{M}}$	Real world (observed) and model response (output) vectors	
$\mathbf{y}_{\mathrm{RI_H}}, \mathbf{y}_{\mathrm{M_H}}$	Horizontal interpolated real world (observed) and model response (output) vectors	
$\mathbf{y}_{\mathrm{RI_V}}, \mathbf{y}_{\mathrm{M_V}}$	Vertical interpolated real world (observed) and model response (output) vectors	
$\mathbf{y}_{\mathrm{RI}}$	Interpolated test data response vector	
a	Hyperbolic tangent bias constant (slope parameter)	
b	Hyperbolic tangent scaling constant	
b	Cylinder diameter or bore	m or mm
b_i, b_k	Neuron bias or processing element bias	
b_0, b_1	Linear regression y-intercept and gradient coefficients	
$b_1, b_2, \ldots, b_n$	Polynomial Coefficients	
p, d, R	Dimension of input space or number of network inputs	
d, d_k	Desired or target output	
e_k	Error signal	
n	Supervised training exemplar number or epoch	
n_c	Number of crank revolutions per power stroke	
n_{train}	Number of multiple training runs	
$\dot{m}_c$	Fuel mass flow rate	$kg.s^{-1}$
$\dot{m}_{ca}$	Fuel actual mass flow rate	$kg.s^{-1}$
p	Pressure	$N.m^{-2}$

Symbol	Description	Dimensional Units
p	Hypothesis testing probability	
q_c	Fuel heating value or specific energy content	MJ. kg^{-1}
R^2	Coefficient of determination	
r	Coefficient of correlation	
r_e	Expansion ratio	
r_v	Compression ratio	
s	Piston stroke	m
s, s_R, s_M	Sample standard deviation	
s^2	Sample variance	
t	Time	s
t	t-value: independent and paired t-test	
$\dot{v}_{ca}$	Fuel actual volumetric flow rate	
w_{kj}	Synaptic weight	
x_i	Input variable	
x_U, x_M	White-Box inputs	
x_i^*	Zero-mean unit variance transformed input variable	
y_U, y_M	White-Box response (outputs)	
y_k, y_{ij}	Processing element or network output	
α	Level of significance or Fuel temperature compensation coefficient	
β	Delta-bar-delta: multiplicative constant	
ε	Input-output mapping error	
ε_a	Acceptable error	
ε_{A_i}	Model to desired response absolute error	%
ε_{AIC}	Model response error – Akaike's Information Criteria	
ε_c	Percentage validation score on performance map contours	%
ε_E	Coefficient of Efficiency	
ε_{E_1}	Adjusted Coefficient of Efficiency	

Symbol	Description	Dimensional Units
$\varepsilon_{\mathrm{IA}}$	Index of Agreement	
$\varepsilon_{\mathrm{IA}_1}$	Modified Index of Agreement	
$\varepsilon_{\mathrm{MS}}$	Model response error - Mean Squared Error	
$\varepsilon_{\mathrm{MSR}}$	Mean squared error normalized to Nearer Specification Limit	
$\varepsilon_{\mathrm{MSR_H}}$	Normalised mean squared error – horizontal interpolated data	
$\varepsilon_{\mathrm{MSR_V}}$	Normalised mean squared error – vertical interpolated data	
ε_{R}	Mean propagated error - test data set	
$\phi_j(\bullet)$	Non-linear or Basis Function	
$\varphi(\bullet)$	Processing element activation function	
η	Learning rate parameter	
η_b	Brake thermal Efficiency	
η_{c}	Fuel conversion efficiency	
η_i	Indicated thermal efficiency	
η_{mech}	Mechanical efficiency	
κ	Delta-bar-delta: additive constant	
μ_i	Population mean of model input variable	
ρ_{ref}	Reference fuel density	$kg.dm^{-3}$
σ_i	Population standard deviation of input variable	
τ	Engine brake torque	N.m
υ_k	Weighted input to processing element activation function	
ω	Crank angular velocity	$rad.\ s^{-1}$
ξ	Error surface cost function	
ξ_{AV}	Cost function average squared error	
ζ	Delta-bar-delta: smoothing constant	

Abbreviations

AIC	Akaike's Information Criteria
CI	Confidence Interval or Compression Ignition
CPA	Change Point Analysis
CUSUM	Cumulative Summing
df	Degrees of Freedom
FNN	Feed-forward neural network
GA	Genetic Algorithm
GPS	Global Positioning System
HMI	Human Machine Interface
NN	Neural Network
NSL	Nearer Specification Limit
M&T	Monitoring and Targeting
MLP	Multilayer Perceptron
MNN	Modular neural network
MSE	Mean squared error
MSER	Mean squared error – normalized to nearer specification limit
PE	Processing element and potential error
SFC	Specific Fuel Consumption
SI	Spark Ignition

Chapter 1

Introduction

1.1 Diesel Engines - A Historical Perspective

In 1878 Rudolf Diesel heard about the poor efficiency of the steam engine in a lecture on thermodynamics given by Carl Linde; the steam engine efficiency at the time was around 12 percent. During this time he studied a theory by the French physicist Carnot, whose cyclic process promised considerably higher energy efficiency in heat engines. The result of this was that Rudolph Diesel was prompted to develop a theory that revolutionized the engines of his day. He imagined an engine in which air is compressed in the cylinder such that there is a large rise in air temperature. When fuel, under pressure, is introduced into the piston chamber containing this air, the fuel is ignited by the high temperature of the air, causing it to ignite and thereby forcing the piston down. The piston force can then be converted via mechanical systems into useful work.

In February 1893, Rudolph Diesel was granted a patent, Patent No. 67207 "Working Method and Design for Combustion Engines". This, together with contracts from Frederick Krupp and Machinenfabrik Augsburg Nurnberg (MAN), Rudolf Diesel began the development and building of working models of an engine. As early as July 1893 a test engine was completed, which could however, not run under its own power. This failure resulted in continued development, and experimentation continued until February 1897 when the first model ran under its own power with 26 percent efficiency; remarkably more than double the efficiency of the steam engines of this period.

Although efficient, the engine was heavy in relation to its power output. The engine weighed 4,500 kg and had a power output of 15 kW. This is a specific weight of 300 kg/kW which is very poor in relation to modern diesel engines that have specific weights in the range 2.5 to 50 kg/kW. One reason for the high specific weight of the engine is the higher pressures that occur in the cylinder, which can only be controlled by a sturdier design. However, the main reason lies in the introduction of the fuel into

the cylinder under high pressure. The only feasible method was to introduce fuel using pressurised air, and this was only possible with auxiliaries that were not only heavy but also part of the engine. This high specific weight meant that practical diesel engines were only applied for stationary use. In the years 1910 to 1922 the stationary diesel engine continued to be used and improved upon and in 1922 Robert Bosch began the development of a fuel injection system for the diesel engine. By 1927 Robert Bosch finally had an acceptable fuel injection pump, leading to a massive improvement in the specific weight of diesel engines, thus opening the field of application to mobile use. The result of these efforts was a massive increase in the versatility of diesel engines in practical applications.

Rudolf Diesel's theory and designs were not able to realise the efficiency levels of the Carnot cycle. However, the diesel engine remains the most economical practical thermal engine. The majority of all land and sea freight, construction, mining, remote/standby power generation and military operational functions are accomplished using equipment powered by diesel engines. Modern diesel engines are highly efficient (35% to 50%) and are currently built to generate powers from less than 1 kW up to 80,000 kW. The diesel engine's versatility, efficiency and power range make it a workhorse throughout industry.

Throughout the industrialised world there is a massive conversion of fuel (chemical energy) into motive energy and electrical energy, and this is typically a significant cost to each economy. This cost is measured in terms of both economic and environmental impact. Due to these costs there is a push towards more efficient use of energy.

1.2 Efficient use of Energy: A Strategic Measure

Economies, and for that matter people, are highly reliant on energy that is sourced by converting fossil fuels into heat and power. Fossil fuels and specifically oil are a diminishing resource. Therefore countries face tremendous challenges with respect to finding alternatives to oil (Roberts, 2004). Although alternative sources of energy are being sought and new cleaner and safer technologies are being developed, global reliance on energy from oil will be the norm for many economies way into the 21st century. In the next decade, there is requirement and necessity to develop energy conservation strategies and enabling technologies for the extension of energy

conversion from oil. Oil can be conserved by efficient use of energy. Energy efficiency is the main strategic measure countries can use, not only for the conservation of fossil energy resources, but also for abatement of air pollution.

The cost savings from environmental improvements and efficient use of energy derived from fuel can provide a competitive edge for manufacturers. This is due to increasing private sector investment in environmentally and socially sustainable development. Sustainable development is good business; many lenders of capital require it, and communities, investors and customers (local and foreign) insist on it. Also, many employee and public health and safety issues are addressed by clean burning and efficient use of fuel. The term efficient *use of energy fuel* includes all technological and economical measures aimed at reducing the specific energy demand of a production system or economic sector requiring a diesel engine as its prime mover.

Diesel engines used as prime movers for power generation, industrial processes and marine propulsion are significant consumers of fuel, hence the economic and environmental cost involved. It is therefore important that these engines convert their fuel energy efficiently. Diesel engine fuel conversion efficiency can be achieved by monitoring the deviation between the actual fuel consumption and the target consumption. Deviations outside the targeted limits highlight the need for corrective action. Corrective action is typically driven by a well established process called energy monitoring and targeting.

1.3 Energy Monitoring and Targeting

Energy monitoring and targeting (M&T) is the collection, interpretation and reporting of information on energy use. Information provision is energy M&T's primary role within general energy management. Energy management's purpose is to measure and maintain performance of equipment and plant, and to locate opportunities for reducing energy consumption and cost. Energy management that encompasses energy M&T has much in common with quality management, statistical process control and data mining; since energy M&T uses many similar techniques.

Energy management is a cyclical process and is typically formalised within an industrial setting; the process being depicted in Figure 1.1. Viability of the process is

dependant on the ability to convert collected data into information. This is due to the fact that information leads to understanding. Understanding leads to effective action with subsequent performance retention or improvement.

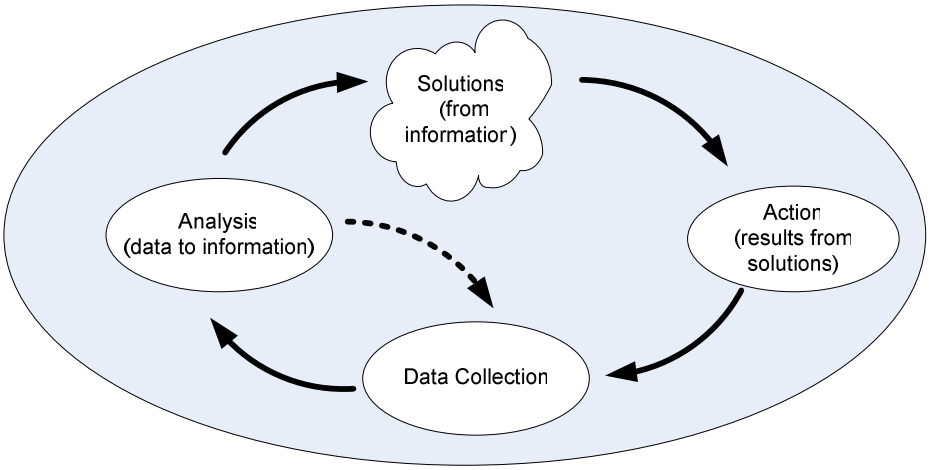


Figure 1.1: The Industrial Energy Management Cycle

Performance improvement via effective action depends on being able to recognise, for the same period, an exceptional deviation between actual energy consumption and the expected or target consumption. Recognition of an exceptional deviation requires, in the context of diesel engine energy M&T, two data streams and an estimating tool (process model) for each consumption period. These three can be defined as:

- the measured fuel consumption or energy usage for the period;
- measurement of the primary operational parameters so that the energy driver, in this case power produced, can be calculated for the period; and
- a tool or model to estimate the target or desired consumption for the period based on the energy drivers.

The difference between the actual consumption and the target consumption, called the consumption deviation, can then be used to indicate whether there has been an exception in fuel consumption for the period. The M&T process is therefore, in conceptual terms, quite simple and answers “what was the consumption versus what should it have been?”

Although conceptually simple, the M&T process has two practical implementation problems. The first practical problem deals with issues around instrumentation for fuel and engine operating parameter measurement and the associated logging systems. This problem is typically not technically constrained and can be addressed by competent technical review. The second practical problem deals with issues around estimating the target consumption. Target consumption estimation (prediction) is the most difficult technical part of M&T since it requires a tested and validated model. Such validated predictive models are either difficult to obtain or too complex for practical implementation. Therefore, operational M&T on diesel engines is typically constrained by the non-availability of practical process models for predicting engine consumption.

1.4 Engine Performance Prediction

The only basis for predicting the specific fuel consumption of an engine is by using an engine performance map (Heywood, 1988; Golverk, 1994a/b; Celik and Arcaklioglu, 2004). An engine performance map is a graphical representation of specific fuel consumption versus engine load and speed. Engine performance maps are used extensively in fuel economy simulation programmes (SAE, 1995) and are valuable in that they allow for the estimation of any particular engine's operational fuel economy for M&T purposes. However, completion of an engine performance map requires a minimum of 400 to 600 test measurements of fuel consumption at various combinations of engine load and speed. These tests are costly, time consuming and require highly skilled personnel and specialised facilities. Therefore, complete engine performance mapping is not frequently done (Golverk 1994a). Further, industrial operators of diesel engines typically do not have the opportunity to do this testing. For them, there is no simple or cost effective way of obtaining a complete performance map of an engine. In this respect it is useful to apply methodologies and develop models that can reduce the effort in producing engine performance maps.

The development of models for diesel engine performance mapping is not novel. Golverk (1994a/b) has proposed the use of a quadratic function, the Universal Performance Mapping Method, for estimating engine fuel consumption performance. This model is also proposed as generalised and therefore can be simply applied to any diesel engine. This is in a similar vein to Harris and Pearce (1990) who propose

using a polynomial with nine coefficients thereby providing a universal model for predicting torque, power and fuel consumption as a function of engine operating parameters. Further, there is the use of multivariate least-squares regression for performance modelling - resulting in a matrix of regression coefficients (Barker, 1982), or a two-stage regression approach (Holliday *et al*. 1998). More recently, Celik and Arcaklioglu (2004) proposed a neural network based model for a specific diesel engine performance prediction.

There are considerable deficiencies in traditional engine performance modelling and testing as highlighted by Barker (1982), Holliday *et al.*(1998) and Golverk (1994a). These deficiencies are rooted mainly in modelling complexity and practical implementation issues. These issues are shown to be addressed by artificial neural networks, which have demonstrated widespread and successful application in many modelling problems. Therefore, a novel approach to engine performance modelling is proposed and is based on the aims of Golverk (1994a) and the use of artificial neural networks similar to Celik and Arcaklioglu (2004). Golverk (1994a) proposes a simple method that is generalised to any specified diesel engine, and Celik and Arcaklioglu (2004) demonstrate artificial neural network implementation of engine performance mapping, albeit only on a specific engine. Thus, the model proposed here is based on the power of artificial neural networks and has generalised application functionality on specified normally aspirated direct injection diesel engines. Such a model has high practical utility value for process engineers and managers that need to control energy consumption using M&T.

1.5 Artificial Neural Networks

Artificial neural network modeling (neural networks) encompasses very sophisticated techniques capable of modeling complex functions and processes. The true power and advantage of neural networks lies in their ability to represent both linear and nonlinear relationships as well as having the capability of learning by example. For processes that have non-linear characteristics such as those found in diesel engine performance modeling, traditional linear models are simply inadequate. In comparison to traditional computing methods, neural networks offer a different way to analyse data and to recognise patterns within that data by being generic non-linear approximators (Anderson *et al.* 1992). They have been used extensively for various

purposes like regression (or function approximation as in engine performance mapping), classification and feature extraction.

A neural network is a collection of basic units (processing elements or transfer functions) that compute a non–linear function of their input. Every input has an assigned weight that determines the impact this input has on the output of the processing element. By interconnecting the correct number of elements in a suitable way and setting the weights to appropriate values, a neural network can approximate any function (or input-output mapping), of linear or non–linear problems. This structure of elements and connections, known as the network's topology (or architecture), together with the weights of the connections, determines the network's final behaviour (Matteucci, 2002). Neural networks cannot solve every problem. Traditional methods may be better. Nevertheless, neural networks, when they are used wisely, usually perform at least as well as the most appropriate traditional method and in many cases significantly better (Tarassenko, 1998). The performance comparison between a traditional computing approach by Golverk (1994a) and the neural network approach is extensively tested and discussed in this study.

A significant problem with neural networks is that the inner workings of many neural networks are "black boxes" and typically deny interpretation or provide reasons for a particular generated result - leading to some purists calling the use of neural networks *voodoo engineering* (Anderson *et al*. 1992). This has not prevented neural networks becoming well established engineering tools and common computational methods for many old and new everyday real life tasks and problems (Kecman, 2000). However, interpretation issues remain a problem that often makes neural networks undesirable for decision support applications (Glorfeld, 1996). Decision-support tools based on conventional expert systems are often preferable to neural networks that deny interpretation.

In engineering applications of neural networks it is fundamental that the behaviour of the model is fully understood from underlying physical principles and heuristics. If a neural network is to be viewed as anything but some form of iterative black box that produces accurate predictions, then some methodology of interpreting the relative importance each input has on the output must be applied (Tao, 2000; Glorfeld, 1996). Here, a *domain expert* methodology is applied whereby extensive effort is placed on defining the parameters that generally exemplify diesel engine performance and selecting these as inputs. Further, the relative importance of each input on the

output is tested as is the relationship or mapping between the input parameter and the neural network response. This testing has been done by applying sensitivity analysis using the variable perturbation methods as presented by Saltelli (2000), Tao (2000), Glorfeld (1996) and the graphical method of Frey and Patil (2001).

To ensure a reasonable level of success in the application of a neural network to an engineering problem such as diesel engine performance modelling, there must be a focus on meeting the minimum requirements for neural network model development.

1.5.1 Requirements for Development of Neural Network Models

The development of neural networks requires a number of specific conditions to be met. The most common among them are:

- a data set containing the information that adequately describes the problem;
- a data set size sufficient for both training and testing the network;
- an understanding of the nature of the problem being modelled so that appropriate decisions on creating the network can be made (architecture, transfer functions and learning methods); and
- an understanding of the development tools.

Once these conditions are met, neural networks offer the opportunity of solving application problems that are difficult using traditional computing or modelling techniques. However, the development process of a neural network is not simply logic. It involves empirical skill and an intuitive feel as to how a network might be designed (Anderson *et al.* 1992; Tarassenko, 1998).

1.6 Engine Performance Modelling

This study is based on the hypothesis that a neural network-based diesel engine performance model can predict the specific fuel consumption performance of any specified normally-aspirated direct-injection diesel engine; within acceptable errors and practical application constraints.

The development of such a model requires four primary issues to be addressed. These issues encompass the required conditions as highlighted by Anderson (1992) and Tarassenko (1998). These four issues are:

- definition of the engine parameters that best describe diesel engine performance;
- selection and partitioning of engine performance data which adequately describe the problem;
- neural network selection and optimisation;
- neural network testing and comparison with non-neural network models.

1.6.1 Engine Parameters Defining Performance

The engine parameters and data that correlate to engine fuel consumption performance need to be established. That is, what information characterises the problem? This question is a primary requirement for the development of a neural network and requires an understanding of the various engine parameters and their effect on fuel consumption. A number of questions highlight the problem at hand:

Does one simply input the power and rotational speed to the model?

Power can be normalised using engine displaced volume. The resulting parameter, called mean effective pressure, is essentially constant over a wide range of engine sizes. Engine speed can be normalised by the piston stroke length, which results in a parameter called mean piston speed. Mean piston speed together with mean effective pressure can make the effect of engine size on performance explicit (Heywood 1985 and Golverk, 1994). Making the effect of engine size explicit may improve the generalisation capability of the model.

What is better when defining engine performance: thermal efficiency or specific fuel consumption?

For engines, there is no universally accepted definition of thermal efficiency (Ferguson *et al.* 2001). Therefore, engineers prefer to use specific fuel consumption to compare engines, since fuel flow can be directly measured

and then normalised to power output. Specific fuel consumption can be used to calculate the engine thermal efficiency, provided the fuel heating value is held constant when comparing engine thermal efficiency.

How does cylinder size influence engine performance?

Basic to the understanding of engine performance is an appreciation of the influence of cylinder size (Taylor, 1985). For example, there are cylinder size effects on friction and heat loss which influence the efficiency and therefore the specific fuel consumption of the engine.

Does the specific power (power per unit of piston area) describe engine performance?

Specific power is the product of the mean effective pressure and mean piston speed. It is essentially a measure of the designer's success in using the available piston area regardless of cylinder size (Heywood 1985).

Investigating these types of questions is required, since, the major assumption supporting the hypothesis is that the normalised non-linear performance maps of different direct injection diesel engines are quantitatively and qualitatively comparable. Without this *comparability* of engines, the research goal of model generalisation capability will not be met and utility of the developed model will be limited. Further, as will be discussed later, neural networks tend to have improved input-output mapping performance if *a priori* information is applied to the data set and network inputs. Therefore engine parameters affecting engine performance and engine performance maps are investigated, detailed and discussed in Chapter 2.

1.6.2 Model Training and Testing – Engine Data

Data is the lifeblood of any neural network model training and testing. Here, the model training and testing data was selected from the performance maps of an available selection of direct injection diesel engines running on distillate (diesel) fuel. The data needed to be selected with particular attention given to issues around model generalisation capability. Loosely, a model is said to generalise well if its prediction error on previously *unseen* data is acceptable. Unseen data in this context

can be referred to as out of training sample data. However, model generalisation cannot be guaranteed since there is a complex (sometimes unsolvable) interaction between the neural network structure, its training methods and the data sets (Haykin, 1994). None the less, there are numerous heuristics that have been investigated and implemented to address the effects the interaction has on generalisation. These heuristics are primarily related to the specification of the data set size and data set partitioning. These two aspects of data requirements are as follows:

With respect to the size of the data set, information theory highlights the prerequisite that the number of examples in the training set should be of the same order as the free parameters in the neural network. Specifying the training set size for good model generalisation capability is supported by sound heuristics; the details of which are discussed in Chapter 4.

With respect to data partitioning, these have been partitioned into training, validation and testing data sets; as proposed by Tarassenko (1998). This data partitioning is similar to the threefold partitioning of the data using the double cross-validation method as proposed by Michie (1994). However, with the double cross-validation method a unique test data set is kept aside for generalisation validation of the model. This test data set is therefore the out of sample or *unseen* data set. This results in a stringent generalisation validation process since the data of an entire engine is removed from the training data set before the training set is partitioned. Since this data is not used during the development of the neural network, it allows for model generalisation capability validation. This data partitioning methodology is detailed in Chapter 4.

Careful attention to training data set size and the partitioning processes cannot compensate for inaccurate and incomplete data or data that do not describe the problem. This is because neural networks cannot magically create information not contained in the data set. Therefore special attention is paid to selecting the data for training and testing.

1.6.3 Neural Network Selection, Training and Optimisation

Since the neural networks developed and tested in this work need to predict engine performance, the network in principle has to perform function approximation or non-linear input-output mapping (regression). Further, the network needs to learn by example or be trained via supervised learning. Training networks for function approximation using supervised learning methods leads to the selection of feed-forward back-propagation network architectures for the engine mapping problem. Feed-forward back-propagation networks can be divided into two sub-classes: non-modular and modular. Modular networks have two primary advantages over their non-modular counterparts: lower complexity and interference avoidance (Boers *et al.* 1990). Both sub-classes of neural networks have been used for solving the modelling problem, the construction details and relative advantages of each having been reviewed.

The development of a neural network usually proceeds by an iterative process of design refinement (Tarassenko, 1998). Therefore, some development decisions such as network topology and learning rate parameters need to be resolved by experimental optimisation. Experimental optimisation of neural networks can be thought of as a search problem, where the search space encompasses all possible network parameters. The objective of the experimental optimisation is to minimise an error function while preserving network generalisation capability. Achieving this objective is normally based on trial and error. However, this requires an intuitive feel that tends to result in an *ad hoc* approach to neural network development. An alternative to this *ad hoc* approach is proposed in this study, whereby an evolutionary methodology (genetic algorithms) is applied to search for the optimal parameters that define the neural network.

Genetic algorithms have proven to be a powerful search tool when the search space is large and when it is not possible to write an analytical form for the error function in such a search space (Tabakov, 2001; Matteucci, 2002; Yao, 1992; Yao, 1999). The use of genetic algorithms with neural networks results in improved model performance. In this work, a genetic algorithm and the associated genetic operators are described for use in this work; the details of which are presented in Chapter 4.

The following neural network shells and statistical software where utilised for the neural network training, parameter optimisation and testing:

- MatLab (Math Works, Inc) – various NN toolboxes and analysis functionality
- NeuroSolutions (Neuro Dimensions, Inc) – neural network shell
- SPSS (Apache Software Foundation) - statistics

1.6.4 Model Verification and Validation

For performance and comparison of neural network models, Flexer (1995) has shown the necessity for statistical evaluation of neural network models. His study showed that, in general, the quality of statistical evaluation of neural networks is rather low and therefore recommends a set of minimum statistical requirements for proper neural network modelling. These requirements and the associated performance measures have been detailed and discussed. The performance measures have been combined with specific methodologies for model verification and validation.

In the context of this research, verification is the process of ensuring that the neural network model has been transformed into a computer model with sufficient accuracy (Davis, 1992), and validation is the process of ensuring that the model is sufficiently accurate for its intended application (Carson, 1986). A fundamental issue with validation is that it is not possible to prove that a model is valid for an application (Robinson, 1999). Validation is primarily an issue of the reliance one can place on the model's predictions when used for real world decision making. Establishing an acceptable level of reliance on the model is not easily resolved and requires that the validation process is more about trying to demonstrate that the model is incorrect using a number of alternative tests. This is based on the strategy that the more times a model cannot be proved incorrect, the higher the confidence in the model's predictive capability.

There are numerous statistical significance tests that can provide a reject or not-reject statement for the null hypothesis. The null hypothesis is a statement of equality, or a statement about the model and the real world system being the same. Since a mathematical model can clearly only ever be an approximation of a real system, a null hypothesis that the model and the system are equal is clearly false.

Therefore, the null hypothesis has limited utility in the context of model validation. It is more useful to ask whether or not the differences between the model and system are significant enough to effect any conclusions derived from the model (Law and Comas, 2001; Robinson, 1999; Hills and Trucano, 1999). This approach to model validation falls within the realm of the white-box and black-box validation methodologies. These methodologies use the alternative hypothesis which is a statement about the model and the real world being acceptably equal in terms of the model's intended purpose. The alternative hypothesis acceptance or rejection statements based on various performance measures are detailed in Chapter 4.

Significant emphasis is placed on the verification and validation of the developed models using the cross-validation and test data sets. Twenty one individual tests are performed on the neural network models and ten on the quadratic model for comparative purposes. The results of these tests are presented and summarised in Chapter 5.

1.7 Application Demonstration

A practical example is used to demonstrate the use of the model in an energy monitoring and targeting application using synthetic operational data. The process of monitoring and targeting using the model to predict the target performance is demonstrated using the method of Change Point Analysis (CPA) as developed by Taylor (2000). This is an effective statistical process and is used to show the engineering utility of the outcomes of this study.

Chapter 2

Diesel Engine Performance

2.1 Introduction

Diesel engines are an important part of transportation and industry throughout the industrialized world, and their use has primarily expanded because of increased pressures for fuel efficiency. The chief advantages of the diesel engine over the petrol engine are its fuel economy and durability.

The piston in a diesel engine compresses air to high pressure and temperature. Fuel, when injected into this compressed air, auto-ignites, releasing its chemical energy. The resulting combustion gases expand, doing work on the piston, and then are exhausted. Diesel engine power output (or load) is controlled by the amount of injected fuel rather than by throttling the air intake as is done on the spark-ignition engines. Since there is no part-load throttling on a diesel engine and together with a higher compression ratio, the diesel engine has a higher efficiency than the spark-ignition engine. However, as the result of no throttling, the diesel engine has poor air utilization. Therefore, a diesel engine requires a larger piston displacement for the same power output as a comparable spark-ignition engine. This factor, combined with the high compression ratio and auto-ignition combustion pressures, requires the structure of a diesel engine generally to be more massive than its spark-ignition counterpart for prolonged and reliable operation.

Diesel engines may be broadly identified as being either two-stroke or four-stroke cycle, direct-injection or indirect-injection, and naturally-aspirated or supercharged. They also are classified according to service requirements such as automotive, industrial, rail or marine engines. This study focuses on developing a model for performance prediction of direct-injection naturally-aspirated four-stroke diesel engines, and in the service classifications: light to heavy duty automotive and small to medium industrial. [note that the modelling methodologies of this study are not limited by the engine service classification and can be generally applied]

Since diesel engines operate under a wide range of speeds and loads, it is useful to describe the engine fuel consumption in terms of these operational parameters. This is accomplished using an engine performance map, where this map is used to estimate the engine fuel consumption over the entire operating cycle.

Diesel engine fuel consumption performance is characterised with several geometric and thermodynamic parameters. In this section, these are reviewed for the purpose of providing domain specific expertise towards the effective selection of the inputs to the neural network models. Input selection based on the understanding of the parameters influencing diesel engine performance leads to improved neural network model generalisation capability; thereby meeting a primary objective of this study.

2.2 Engine Performance Parameters

The engine operational and design parameters characterising naturally aspirated direct injection diesel engine performance are detailed as follows:

2.2.1 Diesel Engine Thermodynamic Cycle

The actual thermodynamic and chemical processes in diesel engines are too complex for complete theoretical analysis. Therefore, it is useful to imagine an idealised process which resembles the real processes and which is simple enough to lend itself to quantitative treatment. Such an idealised process is called the *equivalent air cycle*. The equivalent air cycle, in general, has similar characteristics to the real cycle. For diesel engines, a widely used equivalent air cycle is the limited-pressure or mixed cycle, and is shown by Figure 2.1.

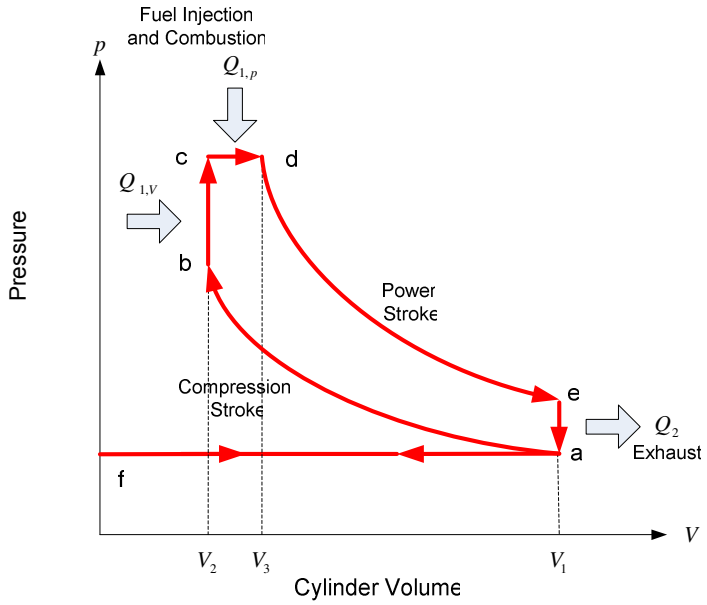


Figure 2.1: Diesel Engine – Ideal Limited-Pressure Cycle

At the start of the cycle, the cylinder contains a volume of air at the pressure and volume indicated by point (a). During the compression stroke, the piston compresses the air reversibly and adiabatically to point (b), thereby raising the temperature of the air. Fuel is then injected and auto ignites, thereby adding heat ($Q_{1,V}$) at constant volume – the process from point (b) to point (c), as well as adding heat ($Q_{1,p}$) at

constant temperature – the process from (c) to (d). Reversible adiabatic expansion of the gas (the power stroke) then takes place to the original volume – the process from point (d) to point (e). Heat (Q_2) is then rejected and the gas returns to its original pressure at constant volume – the process from point (e) to point (a). The gas is then exhausted and a new air charge is taken in at the end of the exhaust, as indicated by the processes (a)-(f)-(a).

The real diesel engine cycle has differences to the ideal limited pressure. Despite the differences, it is informative to use the ideal limited-pressure cycle to define the parameters that characterise engine performance.

2.2.2 Displaced Volume and Volume Ratios

With reference to Figure 2.1, the volume displaced per cylinder is given by:

$$V_{c,d} = V_1 - V_2 \tag{2.1}$$

or

$$V_{c,d} = \frac{1}{4}\pi b^2 s \tag{2.2}$$

where, b is the cylinder diameter or bore (m) and s is the piston stroke length (m).

From equation (2.2), the engine total displaced volume is defined as

$$V_d = \frac{1}{4}\pi b^2 \, s \, n_c \tag{2.3}$$

where, n_c is the total number of engine cylinders.

In the diesel engine, air is compressed adiabatically with a compression ratio defined by

$$r_v = \frac{V_1}{V_2} \tag{2.4}$$

and is typically between 15 and 20. This compression raises the air temperature to typically 800 K to 880 K (Makartchouk, 2002).

A further parameter of interest is the cylinder volume change during combustion, called the expansion ratio, and is defined by

$$r_e = \frac{V_3}{V_2} \tag{2.5}$$

The effect of expansion ratio on cycle efficiency is described later when engine efficiency characteristics are presented.

2.2.3 Power and Mean Effective Pressure

Work Done per Cycle

The area enclosed by the thermodynamic processes of Figure 2.1, is equivalent to the indicated work done per cylinder and per cycle. The indicated work is obtained by integrating around the p-V diagram to obtain the area enclosed (Heywood,1985):

$$W_{c,i} = \oint p\, dV \tag{2.6}$$

Therefore, $W_{c,i}$ is the gross cycle work done during the compression and expansion strokes of the piston.

Indicated Power

Indicated power is the rate of work transfer between the gas in the cylinder and the piston, and can be derived from the indicated work as (Heywood, 1985):

$$\dot{W}_i = \frac{W_{c,i}\, N}{n_c} \tag{2.7}$$

where, N is the crank rotational speed and n_c is the number of crank revolutions for each power stroke per cylinder. The number of crank revolutions for per power stroke is: two for four stroke cycles and one for two stroke cycles.

Brake Power

The net power available at the engine crankshaft is defined as the brake power ($\dot{W}_b$) of the engine. Brake power is less than the indicated power because some of the power from the combustion gas is lost to overcome mechanical losses. Mechanical losses in diesel engines are described as pumping losses (the friction associated with gas transfer), mechanical friction in the rings, bearings, valve train, and other moving parts, losses to run accessories such as the oil pump and water pump, and

windage losses due to rotating parts (Ferguson, 2001). These losses are combined together and, in this study, called friction power ($\dot{W}_f$).

Often it is useful to think of the brake power in terms of the fraction of the indicated power that is available at the engine crankshaft. Therefore, the brake power can be defined, in terms of indicated and friction power, as:

$$\dot{W}_b = \dot{W}_i - \dot{W}_f \tag{2.8}$$

This power is called the brake power of the engine, since it can be calculated from torque and speed measurements taken on a brake dynamometer (or engine performance test rig) at the output shaft of the engine. Therefore, brake power (the rate at which work is done at the crankshaft), can be defined as the product of the torque (the measure of the engine's ability to do work) and the angular velocity of the crankshaft. Therefore brake power is given as:

$$\dot{W}_b = \omega\tau = 2\pi N\tau \tag{2.9}$$

where, N is the angular speed (rev.s^{-1}) of the crankshaft and τ is the torque (N.m) on the crankshaft.

Mean Effective Pressure

While brake torque is a valuable measure of a particular engine's ability to do work, it depends on engine size. The effects of engine size on performance need to be normalised so as to make engines comparable. Therefore, a more useful relative engine performance measure is obtained by dividing the work done per cycle $W_{c,i}$ by the cylinder volume displaced per cycle V_d. The engine parameter so obtained has units of force per unit area and is called the *indicated mean effective pressure*:

$$P_i = \frac{W_{c,i}}{V_d} \tag{2.10}$$

Similarly the *brake mean effective pressure* (P_b) can be defined using equation (2.10) with the brake power ($\dot{W}_b$). Therefore, in this research the *brake mean effective pressure* is defined as:

$$P_b = \frac{2\dot{W}_b}{V_d N} = \frac{4\pi\tau}{V_d} \tag{2.11}$$

where, the units of P_b are kPa or $1x10^3$ $N.m^{-2}$ if V_d is in units of dm^3 and τ is in units of N.m.

The maximum brake mean effective pressure of good engine designs is well established, and is essentially constant over a wide range of engine sizes (Heywood, 1988). For naturally aspirated four stroke diesels, the maximum brake mean effective pressure is in the 700 to 900 kPa range.

The difference between the indicated mean effective pressure and the brake mean effective pressure is the mean effective pressure lost to mechanical losses (friction losses, pumping losses, and engine auxiliary losses). This is defined as the *frictional mean effective pressure* (P_f):

$$P_f = P_i - P_b \tag{2.12}$$

For slow speed engines the average value for P_f is 100 to 180 kPa, for medium speed engines P_f is 170 to 200 kPa, and high speed engines P_f > 200 kPa (Makartchouk, 2002).

The most significant component of P_f is the *mechanical friction mean effective pressure* with the following scaling (Ferguson, 2001):

$$P_{f_m} \sim \frac{F_f U_p}{N V_d} \sim \frac{F_f U_p}{n_c N b^2 s} \tag{2.13}$$

where: F_f is the frictional force, U_p is the piston speed, b is the cylinder diameter (or bore) and s is the piston stroke.

Ferguson (2001) highlights that frictional mean effective pressure increases nearly linearly with piston speed, with no significant variation with load. This relationship is demonstrated in Figure 2.2.

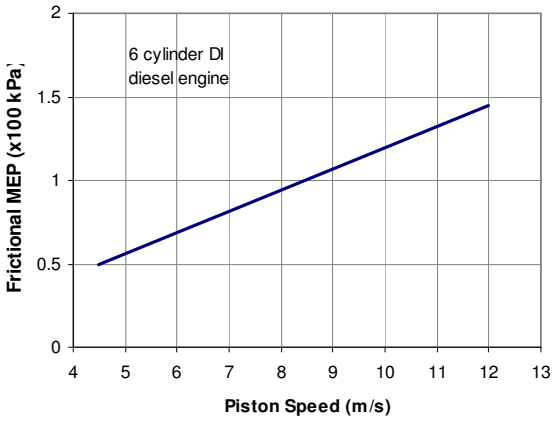


Figure 2.2: Frictional Mean Effective Pressure as a Function of Piston Speed (Ferguson, 2001)

The effects of a small bore and high piston speed on P_{f_m} are clear from equation (2.13). The effects of cylinder size on fuel consumption are detailed and discussed in section 2.3. However, before then it is useful to discuss a further important parameter characterising engine performance: the mean piston speed.

2.2.4 Mean Piston Speed

Mean piston speed is often a more appropriate parameter than crank rotational speed for correlating engine behaviour as a function of speed (Heywood, 1988). Mean piston speed, a function of engine rotational speed (N) and piston stroke (s), is defined as:

$$\bar{U}_p = 2Ns \tag{2.14}$$

The top end of mean piston speed is typically limited to within the range 8 to 15 $m.s^{-1}$ by gas flow resistances (combustion air intake and combustion gas exhaust) and stresses due to inertia forces. For slow, medium and high speed engines, the rated mean piston speeds are respectively, less than 6 $m.s^{-1}$, 6 to 9 $m.s^{-1}$, and greater than 9 $m.s^{-1}$

Taylor (1988) has also shown that, for most diesel engines, the mean piston speed for highest efficiency lies near 6 $m.s^{-1}$.

Using mean effective pressure and mean piston speed, the effect of engine size can be made explicit (Heywood, 1985). This fact is significant since the hypothesis of this research is based on comparing the performances of different engines. Thus, the engine P_b and $\bar{U}_p$ parameters are important in terms of the goals of this research.

Since engine efficiency has a direct impact on fuel consumption (or energy) efficiency it is pertinent to define and review the efficiency parameters of diesel engines.

2.2.5 Efficiency

An understanding of the brake thermal, indicated thermal and mechanical efficiencies combined with frictional mean effective pressure is required to explain the general topology of the engine performance map surface. The primary efficiencies are defined as:

Mechanical Efficiency

The ratio of the brake mean effective pressure (or brake power) to the indicated mean effective pressure (or indicated power) is defined as the mechanical efficiency of the engine (Taylor, 1985):

$$\eta_{\text{mech}} = \frac{\dot{W}_b}{\dot{W}_i} = \frac{P_b}{P_i} = \frac{P_i - P_f}{P_i} \tag{2.15}$$

Examining the above expression, one can see that the mechanical efficiency is a measure of the friction losses. In general, η_{mech} is a function of engine speed and load because the frictional forces scale with shear stresses (hence velocities) and normal forces (hence pressure). Typically, for slow-speed and high-speed engines, the mechanical efficiency is respectively in the ranges 0.80 to 0.83 and 0.75 to 0.78 (Makartchouk, 2002).

Brake Thermal Efficiency

The thermodynamic definition of efficiency of a cyclical process (one in which the gas is returned to the temperature, pressure, and state obtained at the beginning of the

process) does not apply to diesel engines. This is because they operate by burning fuel in, rather than adding heat to the working medium which is never returned to its original state. However, in engine tests the fuel consumption is measured as a mass flow rate ($\dot{m}_c$) and the measured brake torque and engine speed are used to calculate the brake power ($\dot{W}_b$). These two parameters, combined with the known specific energy content or heating value of the fuel (q_c), can be used to conveniently define the brake thermal efficiency of the engine at the specified operating point by:

$$\eta_b = \frac{\dot{W}_b}{\dot{m}_c\, q_c} \tag{2.16}$$

where brake thermal efficiency (also known as the overall efficiency) is the ratio of the brake power $\dot{W}_b$ to the fuel power (the product of fuel mass flow rate $\dot{m}_c$ and the specific energy content of the fuel q_c). Therefore, η_b is a characteristic of the economy of diesel engine operation.

Using equation (2.16), indicated thermal efficiency can be obtained from the indicated power. Hence, brake thermal efficiency can further be defined as the product of the indicated thermal efficiency η_i and the mechanical efficiency η_{mech}:

$$\eta_b = \eta_i . \eta_{\text{mech}} \tag{2.17}$$

A universally accepted definition of thermal efficiency does not exist. Therefore, engine developers and users tend to use a ratio of fuel consumption rate per unit of power. This ratio is called the specific fuel consumption of the engine (Ferguson, 2001; Taylor, 1985) and characterises the fuel economy of diesel engines.

2.2.6 Specific Fuel Consumption

Since the required fuel mass flow rate is a function of power, a more useful parameter than efficiency is the brake specific fuel consumption Q_c (fuel flow rate per unit of power output). It measures how efficiently an engine is using the fuel supplied to produce work.

Specific fuel consumption is given by

$$Q_c = \frac{\dot{m}_c}{\dot{W}_b} = \frac{\dot{m}_c}{2\pi N\tau} \tag{2.18}$$

where $\dot{m}_c$ is the fuel mass flow rate and $\dot{W}_b$ the brake power. The standard units of Q_c is gram per kilowatt hour (g/kW.h). Q_c is a valid measure of efficiency provided q_c is held constant when comparing efficiencies of engines (Ferguson, 2001).

Brake specific fuel consumption is typically used to measure an engine's efficiency, in this case the *fuel conversion efficiency*, which is given by Heywood (1988) as:

$$\eta_c = \frac{3600}{Q_c . q_c} \tag{2.19}$$

where, q_c is the specific energy content or heating value of the fuel.

2.2.6.1 Compression Ratio Effect on Specific Fuel Consumption

Simulations by McAulay (1965) showed that the indicated specific fuel consumption improves at a faster rate with increasing compression ratio than brake specific fuel consumption. This is due to increasing friction and heat losses with increasing compression ratio. In fact, there is an optimum compression ratio due to these effects, and this is the underlying reason why diesel engines have similar designed compression ratios. For diesel engines, the compression ratio (r_v) is typically in the range 15 to 20. Taylor (1985) has also demonstrated that the minor variations in compression ratio have a small effect on direct injection diesel engine efficiency. Therefore compression ratio is not taken as a parameter that generally characterises diesel engine specific fuel consumption.

However, cylinder size has an effect on engine performance. As engine size increases, brake specific fuel consumption decreases and fuel conversion efficiency increases, due to the reduced importance of heat losses and friction (Heywood 1988). Hence, basic to the understanding of engine performance is an appreciation of the influence of cylinder size (Taylor, 1985).

2.3 Cylinder and Engine Size Effects

The greatest differences in cylinder design are caused by differing requirements of various types of service, for example, automotive, marine propulsion and power generation. Diesel engines are built to a wide range of cylinder sizes, and for a given type of service the cylinder designs are quite similar. Therefore, data obtained from diesel engine ratings should tend to reflect the effects of size which have been shown to hold for similar cylinder designs (Taylor, 1985).

2.3.1 Cylinder Size Effects on Engine Rating:

The rated power and speed of an engine are partly a subjective decision, based on the manufacturer's estimate of the engine's performance so as to ensure sufficient reliability and durability. Despite this, and based on the fact that diesel engine cylinder designs are similar, Taylor (1985) has shown that the rated mean effective pressure ($P_{b_{rated}}$), the rate mean piston speed ($\overline{U}_{p_{rated}}$) and rated specific power all tend to fall as cylinder bore (or diameter) increases.

These trends can be explained as follows:

- Small cylinders are generally used in automotive applications where high rated power in proportion to size and weight are important.
- Large cylinders are used only in services in which there is a requirement for reliability, durability and fuel economy. This encourages the designers to specify low ratings in terms of mean effective pressure and piston speed.
- Unless gas temperatures are reduced, increasing cylinder size proportionally increases cylinder component stresses due to operational temperature gradients. This results in design requirements for lower fuel-air ratios and hence lower rated brake mean effective pressure ($P_{b_{rated}}$) as the bore increases.
- Larger cylinders typically result in engine elements being made in bolted constructions that are less rigid (and more stress limited) than smaller cylindered engines. Thus, lower ratings on mean effective pressure and speed are necessary for larger engines.

These explanations account for the fact that rated mean effective pressure ($P_{b_{rated}}$) and rated mean piston speed ($\bar{U}_{p_{rated}}$) diminish as cylinder size increases. Table 2.1 shows the ratio of rated power to piston area and rated power to piston displacement for a large bore engine and small bore engine for engines of the highest rating, mean rating and lowest rating.

Table 2.1: Ratio of Rated Power to Piston Area and Piston Displacement (Taylor, 1985)

	1	2	3	4	5	6	7
		Highest Rating		**Mean Rating**		**Lowest Rating**	
	Bore (mm)	**kW.m^{-2}**	**kW.m^{-3}**	**kW.m^{-2}**	**kW.m^{-3}**	**kW.m^{-2}**	**kW.m^{-3}**
1	737	1 595	1 866	1 387	1 593	1 006	1 138
2	102	3 005	25 030	1 849	15 470	867	7 281
3	**Ratio**						
4	0.14	1.88	13.41	1.33	9.71	0.86	6.40

Notes to Table 2.1:

The ratio of row 3 is calculated by dividing row 2 by row 1.

The bore to stroke ratio is 0.85 in each case.

From Table 2.1, when comparing the ratio of rated power to piston area for large and small bore engines (columns 2, 4 and 6), the ratio varies from 1.88 to 0.86 for the highest engine ratings to lowest engine ratings respectively. The ratio of rated power to piston displacement (column 3, 5 and 7) varies from 13.41 to 6.40 for the highest rating to lowest rating respectively. This indicates that the engine rated power is more nearly proportional to piston area than to piston displacement when comparing engines of different ratings. Therefore, this fact is taken as a general characteristic of diesel engines in this study.

The power per unit piston area, the *specific power*, is a measure of the engine designer's success in using the available piston area regardless of cylinder size (Heywood, 1985). Specific power is proportional to the product of the mean effective pressure and mean piston speed and is defined as:

$$\dot{W}_A = \frac{\dot{W}_b}{A_p} = \frac{\bar{U}_p . P_b}{4} \tag{2.20}$$

where the units are kW.m^{-2} and A_p is the engine piston area. In this vein, Taylor (1985) reviewed the data from numerous diesel engines of varying size and the rated specific power. This resulted in a statement regarding the relationship between the rated specific power and the cylinder bore size, given as:

$$\dot{W}_{A_{\text{rated}}} \cong 4480 \times b^{-0.2} \tag{2.21}$$

The rated specific power as a function of cylinder bore is shown in Figure 2.3 for cylinder bores in the range 50mm to 550 mm.

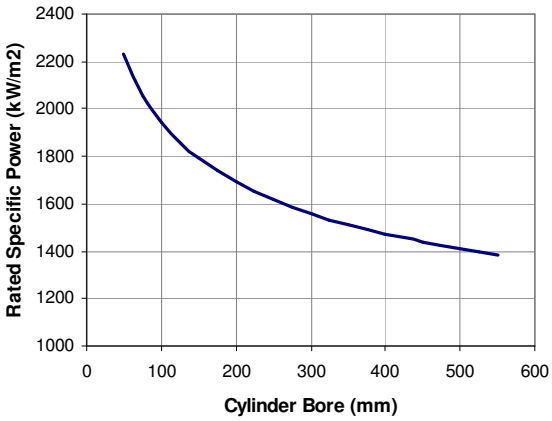


Figure 2.3: Rated Specific Power as a Function of Cylinder Bore

Taylor (1985) has also noted that the lower-rated engine brake mean effective pressure, mean piston speed and specific power ratings are independent of cylinder bore size. This is since the lower engine ratings at each bore size are for services where long life, good reliability and low specific fuel consumption are important. Engines with these service requirements tend to be higher capacity and therefore have larger cylinder bore sizes. Basically, engines with larger bore have lower-rated specific power. This can be clearly seen in Figure 2.3.

Ballany (1998) reviewed the specific fuel consumption performance maps of spark ignition and two and four stroke direct injection diesel engines. The following was noted with respect to specific power:

- The ratio of the specific power at the point of lowest specific fuel consumption ($Q_{c_{\min}}$) to that at the rated point showed remarkable similarity across the different engines; with a range of 0.425 to 0.525.

- The region of lowest specific fuel consumption ($Q_{c_{\min}}$) occurs nearly at the same mean piston speed $\bar{U}_p$ and the same value of the ratio $\frac{\dot{W}_{A_{\min}}}{\dot{W}_{A_{\text{rated}}}}$. The value of mean piston speed corresponding to $Q_{c_{\min}}$ has been previously shown by Taylor (1985) to be approximately 6 m.s^{-1} for most diesel engines.

Since, the bore size for any engine will be readily available and because of the relationship defined by equation (2.21), the rated specific power is used as an input to the model to characterise diesel engines.

2.3.2 Cylinder Size Effects on Efficiency and Rated SFC:

Thomas (1984) has shown that the change in rated specific consumption with change in bore is significant with respect to fuel economy. The nominal rated brake specific fuel consumption as shown in Figure 2.4 can be approximated by

$$Q_{c_{\text{nom rated}}} \cong 570 \times b^{-0.196} \tag{2.22}$$

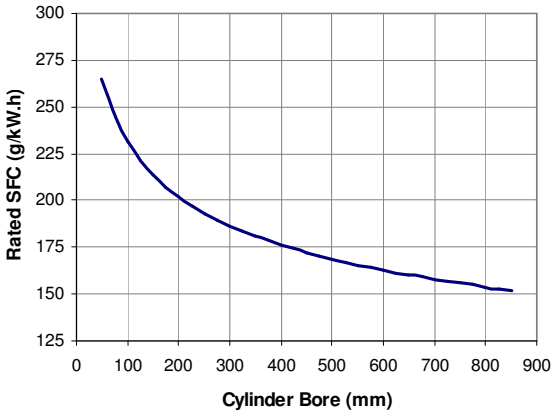


Figure 2.4: Rated Brake Specific Fuel Consumption of Two and Four Stroke Engines vs. Cylinder Bore (Thomas, 1984)

For the modelling in this research, equation (2.22) was modified to include the difference between the rated specific fuel consumption and the minimum specific fuel consumption of the engine. Thus providing a standard specific fuel consumption that is defined as

$$Q_{c_{\text{std}}} = 570 \times b^{-0.196} + \left(Q_{c_{\text{rated}}} - Q_{c_{\text{min}}} \right) \tag{2.23}$$

This function is shown in Figure 2.5, where $\text{Delta SFC} \equiv (Q_{c_{\text{rated}}} - Q_{c_{\text{min}}})$ is plotted in the range 250 to 450 g/kW.h and the cylinder bore size varies from 50 to 150 mm.

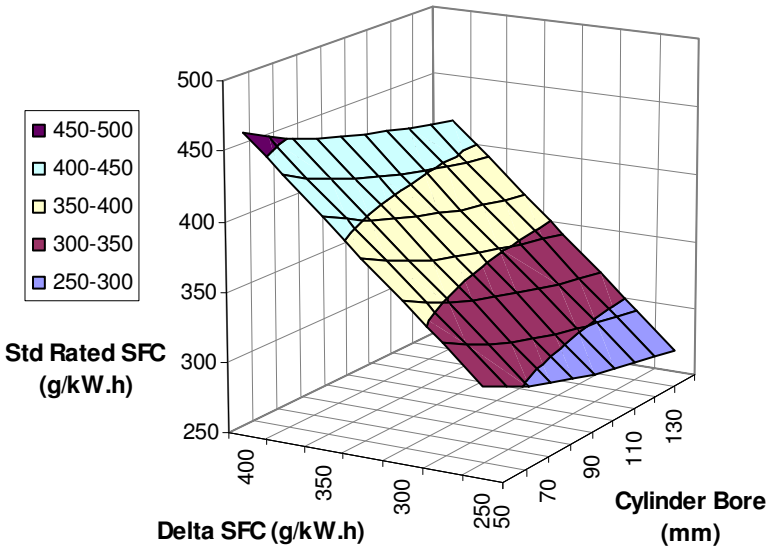


Figure 2.5: Standard Rated Specific Fuel Consumption

The offset can partly be justified, since it has been shown by Ballaney (1998) that there is a fairly constant ratio between the rated point and the point of minimum specific fuel consumption when considering the specific power parameter. Further, there is a need to accommodate for the difference in thermal efficiency between well-designed and poorly-designed engines. That is, in poorly designed engines there may be a significant difference in the minimum specific fuel consumption and rated specific fuel consumption of an engine. Thus, the offset should provide for improved model generalisation capability; thus improved utility. It also will allow for an offset to be input to the model so as to be able to calibrate the model output so as to fit actual in-service engines, if required.

An important factor supporting the trend shown in Figure 2.4 is that the surface to volume ratio of the cylinder is decreasing with increasing bore size, as can be seen by the relationship between the ratio of piston area to volume displaced:

$$R_{AV} = \left(\frac{A_p}{V_d} \right) \sim b^{-1} \tag{2.24}$$

This means that there is reducing heat loss as the cylinder size increases (Ferguson, 2001). A further factor is that the rotational speed typically decreases with increasing bore size:

$$N \sim b^{-1} \tag{2.25}$$

This means that there is less change in volume during the time of combustion – processes (b) to (c) and (c) to (d) of Figure 2.1. It can be shown theoretically and on real engines that cycle efficiency is greatest if heat can be added at constant volume (Ferguson, 2001; Haywood, 1988; Taylor, 1985; Makartchouk, 2002).

The final factor contributing to reducing specific fuel consumption with increasing bore size is that frictional mean effective pressure decreases as engine bore increases (Von Schnurbein, 1981). This can be expected when reviewing equation (2.13). Friction can be expected to be proportionally less in large engines than in small engines. In Figure 2.4 one can see that these effects are significant for small through to medium sized engines (75 mm < cylinder bore < 500 mm); that is, there is a strong relationship between cylinder bore size and rated specific fuel consumption. One can also see that there is a very weak relationship between bore and rated specific fuel consumption between 500 mm and 900 mm (large engines). Taylor (1985) offers a possible explanation for this in that as the cylinders are made larger, the maximum cylinder pressures are reduced and the rates of pressure rise are lowered. Both of these trends tend to reduce cyclic efficiency and therefore offset the smaller relative heat loss of larger cylinders. Further, the brake thermal efficiencies of state-of-the-art large diesel engines (bore > 500 mm) are about 50%. These large engines are reaching the theoretical maximum efficiency and have low losses due to heat transfer, combustion and efficiency (Ferguson, 2001). This tends to limit the reduction in specific fuel consumption in the bore range 500 mm and above.

2.3.3 Model Input Parameters

In summary, the engine parameters of interest for the modelling process are as follows:

Table 2.2: Engine Parameters of Interest

Parameter	Defining Equation
brake mean effective pressure	(2.11)
mean piston speed	(2.14)
specific power	(2.20)
piston area to volume displaced	(2.24)
standard rated specific power	(2.21)
standard rated specific fuel consumption	(2.23)
brake specific fuel consumption	(2.18)

In the section that follows, an engine performance map is defined and the relationship between each parameter and this performance map is described.

2.4 Engine Performance Maps

Engine mapping is the process of modelling non-linear engine output behaviour as a function of adjustable engine-parameters (Barker,1982). The term *engine mapping* is heavily used to describe a whole range of techniques for defining engine performance for a variety of purposes. Therefore, for the purposes of this study, an engine performance map has two specific definitions:

> In terms of real engine data:
> *An engine performance map is a contour plot showing lines of constant specific fuel consumption on the engine load and speed plane.*
>
> In terms of a model:
> *An engine performance map is a computerised model that predicts the engine specific fuel consumption for a number of relevant simulated or measured engine operational and geometric parameters.*

The engine-mapping process typically used in the automotive industry uses large polynomial models fitted to large data sets (parameter estimation). A complete

engine-mapping model can contain over 1000 terms, far too many according to present understanding, and with very little engineering input. There has been relatively little written about engine mapping in the engineering literature. Further, although heavily statistical in nature it has not received much attention in the statistical literature (Holliday, 1998; Grove, 1995). Hence, the present modelling methodologies offer considerable scope for improvement. The first level of improvement needs to address the ease of use and utility of the model. The primary goal of an engine mapping methodology is to provide a good predictive model for a range of engines, rather than a precise close-fitting model limited to a specific engine (Holliday, 1998).

A real engine's performance map is plotted from data obtained via engine tests run in specially equipped dynamometer cells. From these tests, the measured brake specific fuel consumption maps out the specific fuel consumption over the operating range of the engine. For its completion, an engine performance map typically requires a minimum of 400 to 600 measurements of fuel consumption at various combinations of engine load and speed for its completion. These measurements are costly, time consuming and require highly skilled personnel and specialised facilities. Therefore complete engine performance mapping is not frequently done (Golverk, 1994a). Despite the extensive effort needed for their production, engine performance maps are used extensively in fuel economy simulation programmes (SAE, 1995). This is because an engine performance map is valuable, in that it is the only means for estimation of an engine's operational fuel economy when operated over a wide range of loads and speeds (Golverk, 1994a). It is therefore useful to develop modelling methodologies and techniques that can reduce the effort required for producing engine performance maps. This is the main thrust of this study.

2.4.1 Engine Specific Fuel Consumption Performance Map

Performance maps, where constant specific fuel consumption contours are plotted on a graph of brake mean effective pressure versus engine speed, are commonly used to describe the effects of load and speed variations on fuel consumption and, more specifically, at part load conditions (Heywood, 1988; Harris, 1990; Golverk, 1994a/b). Engine performance maps in this form are vital for estimating operational performance as well as monitoring and diagnosing actual engine performance.

Figure 2.6 is an example of a performance map for a diesel engine. The performance map is for a KHD Deutz 1.5 dm^3 three-cylinder four-stroke air cooled naturally-aspirated direct-injection diesel engine (bore of 100mm, stroke of 100mm, model F3L912) – see Appendix A.2 for a technical description of this engine.

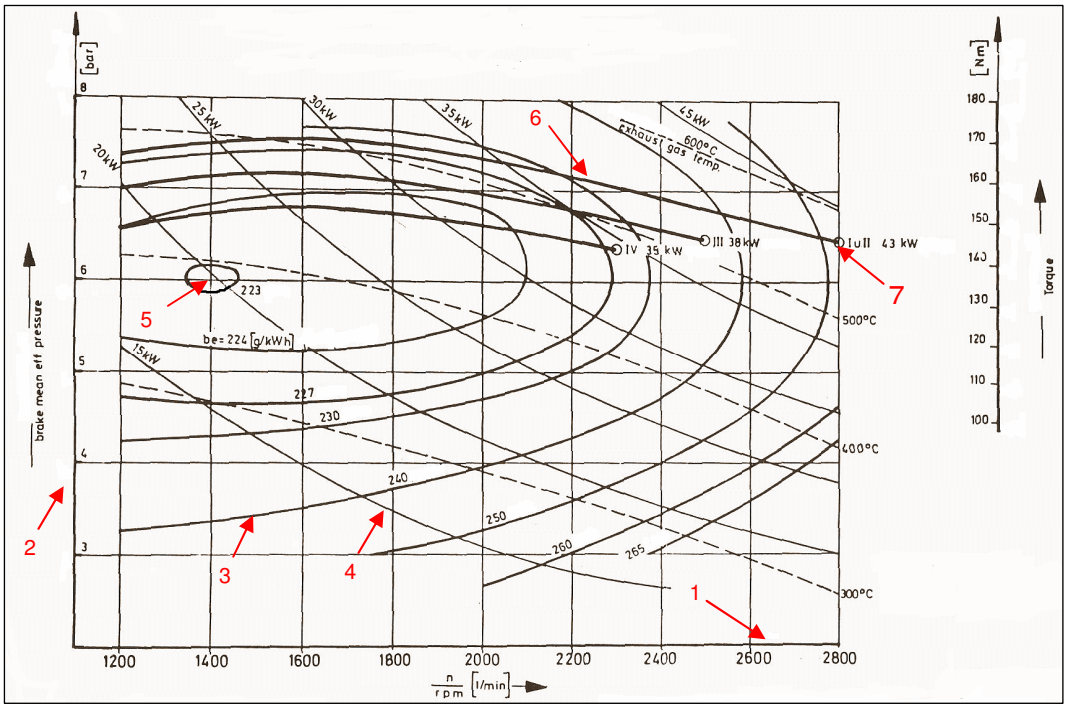


Figure 2.6: Engine Performance Map Example (KHD Deutz)

2.4.1.1 Fuel Consumption Map Features

The features of the engine performance map of interest are as follows (reference numbers are marked on Figure 2.6):

Marker 1: The engine rotational speed axis. The angular rotation of the engine crankshaft (shown in Figure 2.6 as revolutions per minute and dimensioned in equation (2.14) as revolutions per second).

Marker 2: The brake mean effective pressure or torque axis – equation (2.11).

Marker 3: Constant or brake iso-specific fuel consumption contours – equation.(2.18) in units of g/kW.h. These are lines of constant specific fuel consumption in the engine torque and speed plane. Note that these are not lines of constant fuel consumption. Since the engine specific fuel consumption is plotted in terms of two variables, torque

and speed, the map is in fact three dimensional or has a three dimensional surface on the torque speed plane. Hence, contour lines are used so that the specific fuel consumption surface topology can be indicated on a two dimensional graph. This is no different to contour maps used for land survey maps.

Marker 4: The lines of constant power – equation (2.9). On some maps these are the lines of specific power – equation (2.20)

Marker 5: Point or region of minimum brake specific fuel consumption. The region of minimum brake specific fuel consumption is the region where the product of the indicated thermal efficiency and the mechanical efficiency are the highest, that is, where the brake thermal efficiency is the highest – equation (2.17).

Marker 6: Engine rated power curve. This line defines the boundary of the rated torque and speed (or rated power) of the engine. Operation above the line is typically governed by the engine management system.

Marker 7: The engine rated power at the rated speed and rated mean effective pressure or torque. This is the maximum allowable power of the engine for the specified service.

2.4.1.2 Fuel Consumption Performance Map Topology

As summarised using Golverk (1994a), Heywood (1988), Taylor (1985), Makartchouk (2002) and Thomas (1984), the typical topology of the brake specific fuel consumption performance map of Figure 2.6 is due to the following:

2.4.1.2.1 Along a Line of Constant Speed

Moving down from marker (5), the region of lowest brake specific fuel consumption, along a line of constant piston speed, as brake mean effective pressure is reduced, mechanical efficiency drops off. The reduction in mechanical efficiency results in an increase in brake specific fuel consumption. The drop off in mechanical efficiency is due to the indicated mean effective pressure reducing whilst the frictional mean effective pressure essentially remains constant. This can be seen from equation (2.13), where frictional mean effective pressure is proportional to piston speed; which has remained constant. Moving upwards from marker (5) by increasing the brake

mean effective pressure along a line of constant piston speed, the fuel-air ratio increases at such a rate that the indicated thermal efficiency decreases faster than the mechanical efficiency increases. The result is a drop in brake thermal efficiency and an increase in brake specific fuel consumption.

Along a Line of Constant Brake Mean Effective Pressure

Moving right from marker (5) along a line of constant brake mean effective pressure, as piston speed increases, mechanical efficiency decreases while the indicated thermal efficiency remains nearly constant. The result is a decrease in brake thermal efficiency and hence an increase in brake specific fuel consumption. In this case the mechanical efficiency has reduced due to the increasing frictional losses. Now, moving left along a line of constant brake mean effective pressure, as the piston speed reduces, the product of the mechanical efficiency and indicated thermal efficiency tends to remain fairly constant down to the lowest engine operational speed. This is because the decrease in frictional losses (or increasing mechanical efficiency) with reducing piston speed is nearly balanced by a decrease in the indicated thermal efficiency. The result is a marginal increase in brake specific fuel consumption.

When varying the load and speed of the engine, the combined effects of changes in indicated thermal efficiency and mechanical efficiency on brake thermal efficiency result in the distinctive shape of the engine performance map; a stretched bowl shape. The general shape is similar for all normally aspirate direct injection diesel engines, however these are only quantifiably comparable when scaled on the correct coordinates.

2.4.2 Normalised Engine Fuel Consumption Performance Maps

Power and torque depend on an engine's displaced volume or basically the engine size. Hence, the performance maps of diesel engines differ one from another by both scale and shape (Golverk, 1994a). However, using normalised parameters such as mean piston speed and mean effective pressure, the effect of engine size can be made explicit (Heywood, 1988). For different diesel engines, the normalised engine performance maps are similar in scale and general shape. Also, when plotted against the mean piston speed they are quantitatively comparable (Heywood, 1988). This is a critical fact that supports the successful outcome of this study.

2.4.2.1 Golverk Normalisation

Golverk (1994a) proposed that, for the purpose of their generalisation, engine performance maps be plotted in dimensionless coordinates. In such case they are called normalised engine performance maps. With reference to Figure 2.6, with the Golverk normalised engine performance maps, engine mean piston speed is normalised to the rated mean piston speed ($\bar{U}_{P_{\text{rated}}}$ - at marker 7), mean effective pressure is normalised to the rated mean effective pressure ($P_{b_{\text{rated}}}$ - at marker 7), and the specific fuel consumption is normalised to the minimum specific fuel consumption ($Q_{c_{\min}}$ - at marker 5). Therefore the parameters of the normalised performance maps are defined as follows:

Normalised engine mean piston speed (%):

$$X_{\text{N}} = \frac{\bar{U}_p}{\bar{U}_{P_{\text{rated}}}} \times 100 \tag{2.26}$$

Normalised mean effective pressure (%):

$$Y_{\text{N}} = \frac{P_b}{P_{b_{\text{rated}}}} \times 100 \tag{2.27}$$

Normalised specific fuel consumption (%):

$$Z_{\text{N}} = \frac{Q_c}{Q_{c\,\min}} \times 100 \tag{2.28}$$

2.4.2.2 Golverk Universal Engine Fuel Consumption Performance Map

The scales of the normalised performance maps of different diesel engines are the same. Further, their average normalised specific fuel consumption is similar over the area of the map for which the specific fuel consumption is averaged. Therefore, it can be deduced that the normalised performance maps of all diesel engines are similar (Golverk, 1994a). This provides a possibility for the generalisation and development of a universal normalised performance map suitable for a diesel engine of arbitrary power.

Golverk (1994a/b) developed a quadratic polynomial that can be used to predict the normalised brake specific fuel consumption performance of a normally aspirated diesel engine. The quadratic model is defined by:

$$Z_N = b_1 + b_2 \cdot X_N + b_3 \cdot Y_N + b_4 \cdot X_N^2 + b_5 \cdot X_N \cdot Y_N + b_6 \cdot Y_N^2 \qquad (2.29)$$

where: X_N, Y_N, Z_N are defined in equations (2.26), (2.27) and (2.28); and the quadratic coefficients are as given in Table 2.3 below.

Table 2.3: Golverk Universal Map - Quadratic Coefficients

b_1	b_2	b_3	b_4	b_5	b_6
172.28000	-0.70000	-1.03089	0.0064989	-0.0027573	0.0067941

Golverk (1994a) proposed that the model represented by equation (2.29) and the coefficients listed in Table 2.3 represented a universal normalised performance map. Using this model gave the possibility of approximating the fuel consumption performance map of any normally aspirated direct injection diesel engine. The only values that would have to be known are those at the engine rated regime, that is, the mean effective pressure, the mean piston speed, and the minimum specific fuel consumption of the engine.

It is interesting and informative to note that the coefficient b_3 is 47% greater than coefficient b_2. This indicates that specific fuel consumption prediction is more sensitive to changes in mean effective pressure than to changes in mean piston speed. This characteristic is explored later in the research when dealing with model verification and validation.

The normalised fuel consumption performance map defined by equation (2.29) has been used to access the modelling performance of a neural network-based model. This comparison is required to establish whether the neural network-based model results in an improvement in engine performance prediction. In this study, such a comparison is important, since it provides for a benchmark on which to judge the neural network model performance as well as validating the neural network model for the application. This is required, since there is no practical benefit in developing a

new model using an alternative modeling process if the predictive performance of the new model is equal to or inferior to an existing model or methodology.

2.5 Estimation of Total Fuel Consumed

The ultimate goal of the research is to develop a model that allows for the prediction of the total fuel consumed during the operating cycle (duty cycle) of a diesel engine. The previously discussed engine parameters are used to develop a neural network model so that it may be used to calculate the integrated fuel flow rate which is given by Ferguson (2001) as:

$$M_c = \int_0^t \dot{m}_c(t)dt = \frac{A_p}{4}\int_0^t Q_c(t)\,.\,P_b(t)\,.\,\overline{U}_p(t)\,dt \qquad (2.30)$$

Estimation of total fuel consumed (M_c) using equation (2.30) can be accomplished if one has the model for estimating the brake specific fuel consumption (Q_c) as a primary function of the brake mean effective pressure and mean piston speed over time as well as other specified engine parameters. The engine speed and load (torque) are measured and the other engine parameters are known from the engine specification details. Hence, using the model, the brake specific fuel consumption (Q_c) can be estimated for each measured load and speed point of the operating cycle. Note that the model being developed here is not an estimator of M_c but rather an estimator of Q_c.

The utility of equation (2.30) is based on its use as the estimator for the standard fuel consumption of the engine during the operating cycle. The actual fuel consumed in the same operating cycle is measured by calibrated fuel meters. Comparison of the actual fuel consumption to the standard fuel consumption results in a measurement of performance deviation. Any performance deviations outside a set range would indicate changes in engine health that could possibly require intervention. The required intervention would then be determined by performing specific engine tests or investigations. These investigations would normally lead to an understanding of the problem and fixes would then be implemented to restore peak engine condition. The ultimate result is reduced energy wastage and cost. Hence, for any reasonable

engine performance monitoring program, a model of engine fuel consumption is required for estimating the brake specific fuel consumption.

Now that the basis of the an engine performance model has been explained, there is a need to describe, in general, the characteristics, features and minimum requirements of neural networks. In the next chapter, neural networks as a modelling method are reviewed.

2.6 Summary

In this chapter, diesel engine performance has been reviewed so as to provide relevant domain specific expertise to the neural network design process. The primary objective being to establish those parameters that, in general, characterise diesel engine performance maps. This characterisation of diesel engine performance is critical to support the model generalisation capability. The following points highlight the key findings:

- There is no universally accepted definition of engine thermal efficiency. Therefore specific fuel consumption is generally taken as an indicator of engine efficiency. In this work, the neural network has been developed so as to predict the specific fuel consumption of any specified normally-aspirated direct-injection diesel engine.

- In this study, an engine performance map has two specific definitions:

  In terms of real engine data: an engine performance map is a contour plot showing lines of constant specific fuel consumption on the engine load and speed plane.

  In terms of a model: an engine performance map is a computerised model that predicts the engine specific fuel consumption for a number of relevant simulated or measured engine operational and geometric parameters.

- Power and torque depend on an engine's size. Hence performance maps of diesel engines differ in both scale and shape. However, when plotting specific fuel consumption on mean effective pressure and mean piston speed coordinates, the performance maps are quantitatively comparable.

- The distinctive stretched bowl shape of the engine performance map is due to the effects varying load and speed have on mechanical and indicated thermal efficiency, hence brake thermal efficiency.

- Rated mean effective pressure, mean piston speed and specific power all tend to fall as cylinder bore size increases. Further, it has been shown that

rated specific fuel consumption increases with decreasing cylinder bore size. Therefore cylinder size effects have been used to characterise engine performance.

- The engine parameters of interest in this study are:

    - brake mean effective pressure
    - mean piston speed
    - specific power
    - piston area to volume displaced
    - standard rated specific power
    - standard rated specific fuel consumption
    - brake specific fuel consumption

- The quadratic Universal Performance Mapping Method of Golverk (1994a/b) using normalised engine speed, power and specific fuel consumption has been defined so that it can be used to support the neural network verification and validation process.

Chapter 3

Artificial Neural Networks

3.1 Introduction

Artificial Neural Networks (ANN) represent a technology rooted in many disciplines. They are being successfully applied across an extraordinary range of practical problems, in areas as diverse as finance, medicine, engineering, geology and physics. Indeed, anywhere that there are problems of prediction, classification or control, artificial neural networks are being successfully implemented. Their success is due to three unique attributes: universal approximation ability (input-output mapping), the ability to learn from and adapt to their environment, and the ability to invoke weak assumptions about the underlying physical phenomenon (Haykin, 1994). The two most useful properties are:

Nonlinearity on Input-Output Mapping: Artificial neural networks are very sophisticated modeling techniques capable of modeling complex functions and processes. The true power and advantage of ANN lies in their ability to represent both linear and nonlinear relationships. Since they are generic nonlinear function approximators, they offer a different way to analyse data and to recognise patterns within that data (Anderson, 1992). Traditional linear models are simply inadequate when it comes to modelling data that contains non-linear characteristics. Furthermore, ANN keep in check the *curse of dimensionality* problem that affects traditional attempts to model nonlinear functions with large numbers of variables – as is the case with traditional engine mapping models.

Learning by Example: Artificial neural networks *learn by example*. The model developer collects representative data, and then uses *training algorithms* to automatically learn the structure of the data. Use of real data with an appropriate training algorithm is supervised or active learning using an external *teacher*. In conceptual terms, the teacher may be thought of as having knowledge of the external environment that is represented by a set of

input-output examples. The teacher, by virtue of built-in knowledge, is able to provide the neural network with a desired or target response to its inputs. During training, the network, to which the environment is unknown, generates a response to the inputs. The network response is compared to the desired response, thereby generating an error which is used to adjust the network parameters. The aim of the parameter adjustment is to eventually have the network emulate the teacher.

In essence, a neural network is a collection of basic units, neurons, computing a nonlinear function of their input. Every input has an assigned weight that determines the impact this input has on the output of the neuron. By interconnecting the correct number of neurons in a suitable way and setting the weights to appropriate values, a neural network can approximate any function, linear or nonlinear. This structure of neurons and connections, known as the network's *topology*, together with the weights of the connections, determines the network's final behaviour (Matteucci, 2002).

3.1.1 Historical Perspective

The historical perspective presented here is largely sourced from Anderson (1992), Widrow and Lehr (1990), Bishop (1994), Bishop (1996), Tarassenko (1998) and Haykin (1994).

The first step towards artificial neural networks came in 1943 when Warren McCulloch, a neurophysiologist, and a mathematician, Walter Pitts, wrote a paper on how neurons might work by modelling a simple network with electrical circuits. Donald Hebb in his book *The Organization of Behaviour* pointed out in 1949 that neural pathways are strengthened each time they are used - thereby reinforcing the concept on neurons and how they work. At the time that computers were in their infancy in the 1950's, Nathanial Rochester led the first unsuccessful attempt to simulate neural networks at IBM research laboratories. Later attempts were successful; however, emphasis in computing was beginning to dominate, leaving neural research in the background.

The Dartmouth Summer Research Project on Artificial Intelligence in 1956 provided new stimulus for work in both artificial intelligence and neural networks. In the following years, John von Neumann imitated simple neuron functions using telegraph relays and vacuum tubes. At this time, Frank Rosenblatt of Cornell began work on the Perceptron through his intrigue on how a fly processes vision within its eye. The Perceptron was developed and built into hardware and is the oldest neural network still in use today. The single-layer perceptron was found to be useful in classifying a continuous -valued set of inputs into one of two classes. The perceptron computes the weighted sum of the inputs, subtracts a threshold, and passes a result of one of two possibilities. Unfortunately, the perceptron is limited and was proven as such in the book *Perceptrons* by Marvin Minsky and Seymour Papert in 1969.

In 1959, the Multiple ADAptive LINear Elements models were developed by Benard Widrow and Marcian Hoff at Stanford. MADALINE was the first neural network to be applied to practical problems. This neural network is still in commercial use as an adaptive filter for eliminating phone line echoes. Earlier successes with neural networks caused exaggerated claims as to the potential of neural networks. Unmet promises based on outrageous claims caused respected voices to criticise neural network research. This resulted in dormancy of neural network research until the end of 1981.

In 1982 John Hopfield of Caltech showed, with clarity and mathematical analysis, how neural networks could work and what they could do. Hopfield's approach was to create useful devices as distinct from understanding the brain. This relates to current thinking that neural computing is about machines, not brains. At the same time, the US-Japan Joint Conference on Cooperative/Competitive Neural Networks resulted in the US once again funding neural network research (mainly due to fear of being left behind). In 1985 the American Institute of Physics started the annual meeting - Neural Networks for Computing. In 1987, the Institute of Electrical and Electronic Engineers' (IEEE) first International Conference on Neural Networks was held, thereby bringing neural networks and their applications into active use.

Recently there has been a dramatic growth in the number of large scale applications of neural networks in areas as diverse as handwritten signature verification, carcass grading, medical screening and character recognition. Applications in physics include control of mirror segments for telescopes with adaptive optics and second level triggering in high energy physics experiments. During the same period there

have also been many important developments in the theoretical basis for neural computing. These have been accompanied by a shift of emphasis away from the biological metaphor of the 1980's towards a more principled view of neural networks during the 1990's. Currently there are several different communities studying the theoretical foundations of neural networks. This is since, although many of the newest networks are statistically accurate, they do not solve problems absolutely.

A fundamental problem with neural networks is that the inner workings are difficult to interpret, or the reasons for a particular generated result are difficult to access. These issues can constrain the applications of neural networks for engineering decision support. However, neural networks are well established engineering tools and are becoming common computational means for many old and new everyday real-life tasks and problems (Kecman, 2000). Neural networks cannot solve every problem. Traditional methods may be better. Nevertheless, neural networks, when they are used wisely, usually perform at least as well as the most appropriate traditional method and in some case significantly better on the same problem.

3.2 Conditions for Neural Network Applications

Since neural networks learn by example, they cannot solve problems absolutely. Thus, even when a network has been developed, there is no way to ensure that the network is the optimal network. Therefore, their development exacts a number of specific conditions that include (Anderson, 1992; Tarassenko, 1998):

- an understanding of the nature of the problem being modelled so that decisions on creating the network can be made (architecture, transfer functions and learning methods);
- a data set which includes the information which can characterise the problem;
- an adequately sized data set to both train and test the network;
- an understanding of the development and modelling tools.

Once these specific conditions are met, neural networks offer the opportunity of solving problems that are difficult using traditional computing or expert systems.

3.2.1 Understanding the Problem

An important requirement for the use of a neural network is that that there is a known (or strongly suspected) relationship between the proposed known inputs and unknown outputs. This relationship may be noisy but it must exist. Neural networks, like other modelling techniques, should not be applied to engineering decision support problems without a fundamental understanding of the physical relationships between the inputs and expected outputs. Understanding of the input-output relationships typically comes from domain experts or domain specific expertise. This understanding has been developed in Chapter 2.

3.2.2 Network Training Information and Data Set Size

Data is the lifeblood of any neural network application development. This is because neural networks learn from data. However, it is to be understood that neural networks cannot magically create information that is not contained in the training data. Therefore, the training and testing data sets must be selected with insight into the problem so that they contain information characterising the problem. Also, since they are trained non-parametrically, typically a neural network requires a lot of data. Therefore, the training data sets should contain sufficient exemplars to satisfy the heuristically determined conditions for neural network training. The data requirements for this study are detailed in Chapter 4.

3.2.3 Development Tools: Network Categories and Architectures

The majority of the variations between neural network architectures is due to the various learning rules and how those rules modify the network's topology. Most applications of neural networks fall into five primary categories:

- prediction (regression)
- classification
- data association
- data conceptualisation
- data filtering

Table 3.1 shows the differences between the five primary network categories and shows typical architectures and applications for each category (Anderson, 1992).

Table 3.1: Network Category Architecture and Typical Application

Network Category	**Architecture**	**Typical Use**
Prediction (regression) Function Approximation	Feedforward Backpropagation Direct random search Self Organising Map into Back-propagation General Regression NN Mixture density networks	Use input to predict some output (typical regression type problems)
Classification	Learning vector quantization Counter Propagation Probabilistic neural network Mixture Density networks	Use input values to determine the classification
Data association	Hopfield Boltzmann Machine Bidirectional associative memory Spatio-temporal pattern recognition	Used as in classification networks but also recognise data containing errors
Data conceptualization	Adaptive resonance network Self Organising Feature Map	Analyse inputs so that grouping relationships can be inferred.
Data filtering	Recirculation	Smooth an input signal

Although Table 3.1 is not exhaustive in its category, architecture and typical use listing, it is clear that there are many network architectures available for a diverse range of uses. However, when developing models for engine performance prediction, the interest is with architectures that can model continuous functions of input variables. Thus, the obvious network category is prediction/function approximation. In this category, the feedforward back-propagation network is the most popular due to its relatively simple implementation and it has been successfully used to solve many types of problems in a range of engineering applications. Therefore, for the problem of this study, the feedforward backpropagation network architecture was chosen.

The engine performance mapping problem can be thought of as a nonlinear regression problem. In the next section, nonlinear regression is briefly explored so as to give insight into the neural network architecture and functionality.

3.3 Nonlinear Regression

For many practical regression applications there is a need to consider general classes of function. Therefore, representations are sought for nonlinear models that can approximate any given input-output relationship (or mapping) to arbitrary accuracy. One way to achieve this is to use a set of M nonlinear or *basis functions* $\phi_j(\mathbf{x})$ where $j = 1,\ldots, M$, and $\mathbf{x}$ is the input vector. One can then to form a linear combination of these functions, so that (Jordan, 1996):

$$y_k(\mathbf{x}) = \sum_{j=1}^{M} W_{kj}\phi_j(\mathbf{x}) \tag{3.1}$$

where: $y_k(\mathbf{x})$ is the response or output and W_{kj} a weight parameter or coefficient.

For a sufficiently large value of *M*, and for a suitable choice of the basis function $\phi_j(\mathbf{x})$, such a model has the desired universal approximation properties. A familiar example, for the case of one-dimensional input spaces, is the simple polynomial, for which the $\phi_j(\mathbf{x})$ are simply successive powers of the inputs $\mathbf{x}$ and the W_{kj} are the polynomial coefficients. There are a variety of families of functions in one dimension that can approximate any continuous function to arbitrary accuracy. There is, however, an important problem with this approach to universal approximation. This problem is called the curse of dimensionality.

3.3.1 The Curse of Dimensionality

If, for example, one considers an M^{th} order polynomial, then the number of independent coefficients grows as d^M, where d is the dimension of the input space or number of inputs to the model (Jordan, 1996). Thus, for a typical medium scale application with, say 6 inputs, a fourth order polynomial would have approximately 1,300 adjustable parameters. A large number of adjustable parameters is a serious problem for methods that have a power law (or exponential growth in the number of parameters), since, in order to achieve good generalization it is important to have significantly more training data points than adaptive parameters in the model; significantly more meaning at least ten times. The resulting practical problem is that

the model developer simply has insufficient training data. This is exactly the problem faced by many of the traditional engine performance mapping approaches as highlighted in section 2.4.

A solution to the problem lies in the fact that, for most real world data sets, there are strong (often nonlinear) correlations between the input variables such that the data does not uniformly fill the input space but is effectively confined to a sub-space whose dimensionality is called the intrinsic dimensionality of the data (Bishop, 1996). One can take advantage of this phenomenon by considering again a model of the form in equation (3.1) but in which the basis functions $\phi_j(\mathbf{x})$ are adaptive so that they themselves contain weight parameters whose values can be adjusted in the light of the training data set.

Different models result from different choices for the basis functions, and here one can consider the most common model, the multilayer perceptron (MLP). The multilayer perceptron is obtained by choosing the basis functions to be nonlinear logistic functions. This leads to a multivariate nonlinear function (model) that can be expressed in the form:

$$y_k(\mathbf{x}) = \sum_{j=1}^{M} W_{kj}\,\varphi(\bullet) + W_{k0} \tag{3.2}$$

where, W_{k0} is a bias parameter, and the basis functions $\varphi(\bullet)$ are called transfer or activation functions. One form of the activation function is the logistic sigmoid function of the form:

$$\varphi(\bullet) = \frac{1}{1+e^{-n}} \tag{3.3}$$

and, n is given by

$$n = \sum_{j=1}^{d} W_{ji}x_i + W_{j0} \tag{3.4}$$

where, W_{j0} is a bias parameter.

Note that the use of a nonlinear activation (or transfer) function is crucial, since, if $\varphi(\bullet)$ were replaced by the identity, the network would reduce to several successive linear transformations which would itself be linear and therefore useless for nonlinear function approximation.

The multivariate nonlinear model of equation (3.2) can also be represented as a network diagram as shown in Figure 3.1.

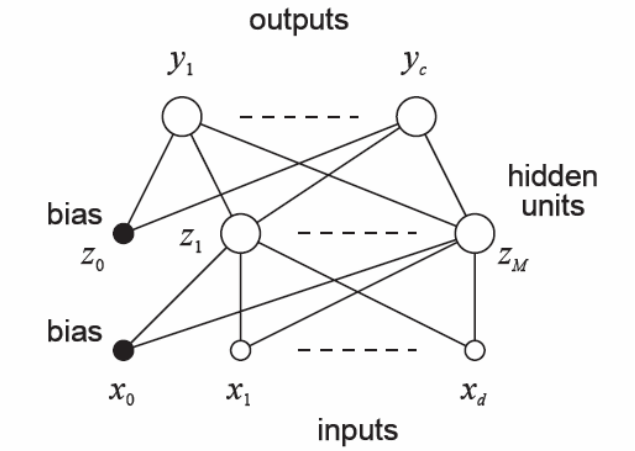


Figure 3.1: Example network having two layers of adaptive weights (Bishop, 1996)

Such a model is able to take account of the intrinsic dimensionality of the data because the first layer weights W_{ji} can adapt and hence orient the surfaces along which the basis function response is constant. It has been demonstrated that models of this form can approximate to arbitrary accuracy any continuous function, provided the number of M hidden units is sufficiently large (Jordan, 1996).

The MLP model can be extended by considering several successive layers of weights. The synonym for such a model with a layer of weights is multilayer feed-forward network.

3.4 Multilayer Feed-Forward Networks

Multilayer feed-forward networks have three distinct characteristics:

- they have hidden units that include nonlinear activation functions that are smooth (or differentiable everywhere);
- they contain one or more layers of hidden units that are not part of the input or output of the network;
- they have a high degree of connectivity between the units.

The combination of these characteristics, together with the ability to learn from experience through training, imparts significant computing power to multilayer feed-

forward networks. These characteristics, however, are also responsible for the deficiencies in understanding the behavior of the network. The use of hidden units with a high degree of connectivity makes the learning process difficult to visualize and comprehend.

It has been shown (Cybenko, 1989; Hornik, 1989) that a two-layer feed-forward network with sigmoid nonlinearity can approximate any function with arbitrary accuracy. However, this is an existence theorem in that it only provides a mathematical justification for the approximation of an arbitrary continuous function. It does not address the practical implementation issues. One would be incorrect to assume that by simply defining the network architecture and choosing the number of hidden units, the network will find a useful solution to the problem. This is because network usefulness is related to its *generalisation* capability. In nearly all engineering applications of prediction type problems (or function approximation), one is interested in the performance of the trained network on previously *unseen* inputs (or generalisation capability). Generalisation is an issue of model order selection and depends on the complexity of the underlying function or process that the neural network is approximating (Tarassenko, 1998). Too many hidden units and over-fitting to the training data will occur; which results in poor generalisation on the unseen inputs or a new data set of interest. The converse is true for too few hidden units (see section 4.2.1 where generalisation issues are highlighted and discussed).

In this type of network, the training or weight optimisation is typically achieved with the highly popular algorithm known as the error back-propagation algorithm (Werbos, 1974; Rumelhart, 1986; Parker, 1985 and LeCun, 1988); which is based on the error-correction learning.

3.5 Backpropagation and Supervised Learning

Backpropagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions (see Widrow and Lehr, 1990; Haykin, 1994; and Rumelhart, 1985 for details). The back-propagation algorithm is used to train a network in a supervised manner by using example training input vectors and corresponding target vectors until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way.

In supervised learning, the difference between the network current output and the desired output is calculated. This raw error is then transformed by the error function (or cost function) to match a particular network's architecture. The most basic architectures use this error directly, some square the error while retaining its sign, some cube the error, and some modify the raw error by the negative logarithm of the likelihood; depending on the specific purposes (Bishop, 1994). This current error is propagated backwards to a previous layer. Hence, the term backpropagation is used in naming this type of learning. The back-propagated value can be either the current error, the current error scaled in some manner (often by the derivative of the activation function), or some other desired value depending on the network type. Normally, the back-propagated value, after being scaled by a *learning rate* parameter, is multiplied against each of the incoming connection weights to modify them before the next learning cycle. The value of the parameter defining the rate of learning (or weight update) has significant influence on the network's ability to converge on a solution to the problem and on the network's stability during training.

Each learning cycle, commonly known as an *epoch* is complete when the entire set of training examples has been presented to the network. If weight updating is performed after the presentation of all the training examples that constitute an epoch, the training is said to be *batch mode* training. Batch mode training has specific beneficial properties that allow for adaptation of the learning rate parameters of the network. These are explored in Chapter 4.

Multilayer feed-forward networks, when trained in a supervised manner, have been applied successfully to solve some difficult and diverse problems. They are, therefore, an important class of neural network and are subsequently reviewed in terms of the application of this research. This review requires that the principle components comprising a multilayer feed-forward network be defined, so that the methodology presented and discussed in Chapter 4 has a foundation.

3.6 Components of a Multilayer Feed-forward ANN

There are seven important components which make up the structure of multilayer feedforward networks. These are summarised and briefly reviewed here in order to support discussion on the development and testing methodology.

3.6.1 Neuron or Processing Element (PE)

Figure 3.2 shows a nonlinear model of a neuron or processing element (Haykin, 1994) and is fundamental to the operation of a neural network. Note that this model is mathematically identical to equation(3.2).

The neuron or processing element (PE) is made up of weighting factors and bias terms, summing function and activation function. These components are valid whether the neuron is used for input, output, or is in one of the hidden layers.

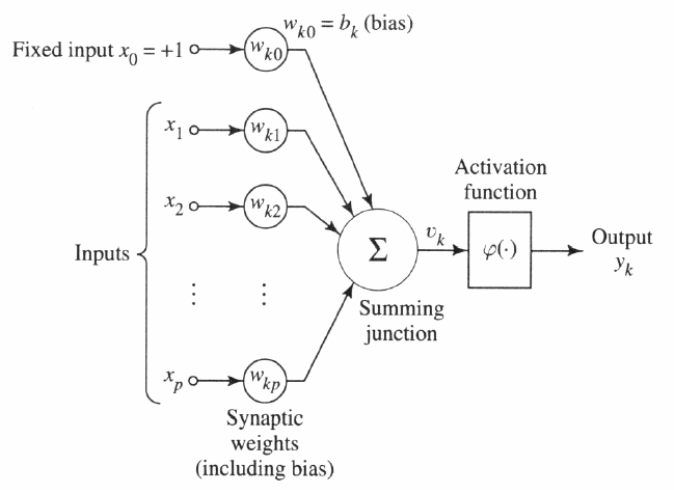


Figure 3.2: Nonlinear Model of a Network PE – Haykin (1994)

3.6.1.1 Weighting Factors (W_{kj})

A neuron usually receives many simultaneous inputs ($x_1, x_2, \ldots, x_p$). Each input connection has its own relative weight ($w_{k1}, w_{k2}, \ldots, w_{kp}$) which gives the input the influence that it needs on the processing element's summation function. These weights make some inputs more important than others so that they have a greater effect on the processing element as they combine to produce a response. The weights are the adaptive coefficients within the network that determine the intensity of the input signal as produced by the neuron in response to the inputs. Therefore, they are a measure of an input's connection strength. The weight strength is positive if the associated connection is excitatory and negative if the connection is inhibitory. These strengths can be modified in response to various training sets and according

to a network's specific topology or through its learning rules. This aspect of neural network development is discussed in Chapter 4.

3.6.1.2 Bias Term (b_k)

Sigmoid-based processing elements in the hidden and output layers usually use a *bias* or threshold term in computing the net input to the processing element. For a linear-based processing element on the network output, a bias term is equivalent to an intercept in a regression model. A bias term can be treated as a connection weight from a special unit with a constant, nonzero activation value and the bias term value can be learned as per the other weights.

The bias term has the following benefit: Consider a multilayer perceptron with a sigmoid activation function, and say there are p inputs to that processing element which define a p-dimensional input space. Any given processing unit draws a hyper-plane through that space. The weights determine where this hyper-plane lies in the input space. Without a bias term, this hyper-plane is constrained to pass through the origin of the space defined by the inputs. For many problems the hyper-plane would be much more useful somewhere else than the origin.

The *universal approximation* property of multilayer perceptrons with sigmoid-based hidden layer activation functions does not hold if one omits the bias terms (Hornik, 1993). Also, networks without output biases are usually ill-conditioned and harder to train than networks that use output biases (van der Smagt, 1998). Hence biases have been used in the networks developed here.

3.6.1.3 Summation Function

The first step in a processing element's operation is to compute the weighted sum of all of the inputs. The inputs and the corresponding weights are vectors which can be represented as $[x_1, x_2, \ldots, x_p]$ and $[w_{k1}, w_{k2}, \ldots, w_{kp}]$. The total input signal is the dot, or inner, product of these two vectors.

This simplistic summation function is found by multiplying each component of the input vector by the corresponding component of the weight vector and then adding up all the products, that is:

$$\upsilon_k = \sum_{j=0}^{p} w_{kj} x_j \tag{3.5}$$

The result (υ_k) is a scalar, not a multi-element vector. Geometrically, the inner product of two vectors can be considered a measure of their similarity. If the vectors point in the same direction, the inner product is maximum; if the vectors point in opposite directions (180 degrees out of phase), their inner product is minimum.
The summation function can be more complex than just the simple input and weight sum of products. The input and weighting coefficients can be combined in many different ways before passing on to the transfer function. In addition to a simple product summing, the summation function can select the minimum, maximum, majority, product, or several normalizing algorithms. The specific algorithm for combining neural inputs is determined by the chosen network architecture. Here the summing of the dot product (or the weighted sum) was applied.

3.6.1.4 Activation or Transfer Function $\varphi(\bullet)$

The activation function, denoted by $\varphi(\bullet)$, defines the output of a neuron in terms of the activity level at its input. The result of the summation function (υ_k), almost always the weighted sum, is transformed to a working output (y_k) through an activation or transfer function.

$$y_k = \varphi(\upsilon_k) \tag{3.6}$$

The most common form of activation function in use is the sigmoid function. A sigmoid function is a strictly increasing function that exhibits smoothness and asymptotic properties. An example of the sigmoid is the logistic function, defined as:

$$y_k = \varphi(\upsilon_k) = \frac{1}{1 + e^{(-a\,\upsilon)}} \tag{3.7}$$

where a is the slope parameter of the sigmoid function, that allows for functions of varying slopes as shown in Figure 3.3 (Haykin, 1994). A critically important property of sigmoid activation function, specifically in regression problems, is that both the function and its derivatives are continuous.

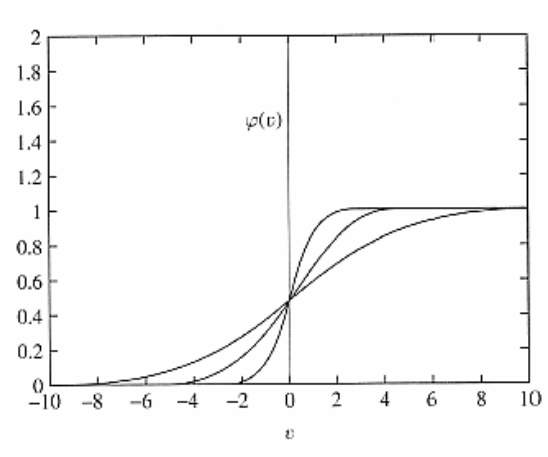


Figure 3.3: Sigmoid function (Haykin, 1994)

Russo (1991) and Guyon (1991) have shown that a multilayer perceptron trained with the backpropagation algorithm, may learn faster if the sigmoid activation function built into the processing elements of the network are asymmetric rather than nonsymmetric. This condition is clearly not satisfied by the logistic function of equation (3.7) and Figure 3.3. Asymmetry can be obtained in the activation function by using a popular form of the sigmoid nonlinearity, namely, the hyperbolic tangent, defined by:

$$\varphi(v) = a\tanh(bv) \tag{3.8}$$

where a and b are constants.

It can be shown that the hyperbolic tangent, as seen in Figure 3.4, is simply the logistic function that has been biased and rescaled.

The hyperbolic tangent is given as

$$a\tanh(bv) = a\left(\frac{1-e^{-bv}}{1+e^{-bv}}\right) = \frac{2a}{1+e^{-bv}} - a \tag{3.9}$$

As a first try, suitable values for the constants can be set as $a = 1.716$ and $b = 0.667$ (Guyon, 1991).

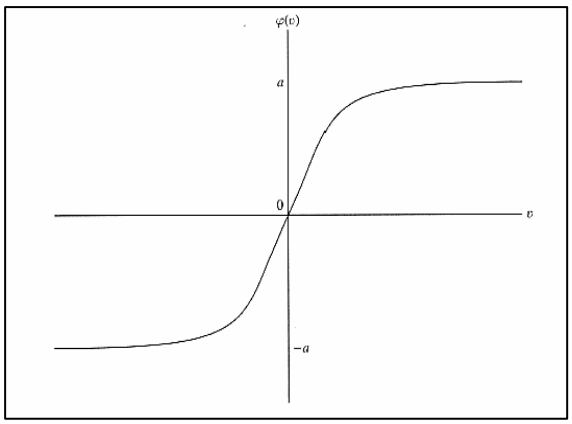


Figure 3.4: Asymmetric Activation Function (Hyperbolic Tangent) – Russo (1991)

3.6.2 Output Function

In this research, the neural network models have used the hyperbolic tangent activation function for the processing elements in the hidden layers. If the last layer of a multilayer network has sigmoid type activation functions, then the outputs of the network are limited to a small range. For the output units, an activation function suited to the distribution of the target values should be chosen. Therefore, for continuous-valued targets with no known bounds, the identity or *linear* activation function (which amounts to no activation function) should be used. This heuristic should be applied unless there is a very good reason to do otherwise (Jordan, 1995). Therefore, in this research the output layer processing elements use the linear combiner or pure linear activation function which is defined as:

$$\varphi(\upsilon) = \upsilon \tag{3.10}$$

3.6.3 Alternative Learning Algorithms

The purpose of the learning algorithm is to adapt the connection weights on the inputs of each processing element. Standard backpropagation is a gradient descent algorithm, in which the network weights are moved (adapted) along the negative of the gradient of the error cost function using a learning rate. There numerous variations on the standard backpropagation algorithm. These variations in algorithm can be distinguished as a global adaptive type, which adapts a single learning rate

for all the connections during training, and the local adaptive type, which use independent learning rates for each connection (Schiffmann, 1993). Most of these algorithms require the batch mode of training.

There are four heuristics underpinning an increased rate of convergence to a solution (see section 4.6.4). There exists several possible implementations of these heuristics; common among them are momentum and the Delta-Bar-Delta learning rule of Jacobs (1988). These are explored in the next chapter.

In the next chapter, the methodology supporting the development of the specific NN of this study is detailed, based on the foundations of this Chapter.

3.7 Summary

In this chapter, the primary features of neural networks have been presented with the emphasis on the components that constitute a feedforward backpropagation network used for nonlinear function approximation. Further, the chapter has laid the foundation for consistent discussion in the later chapters using the terminology associated with neural networks. The following is a point summary of the pertinent issues:

- Feed-forward neural networks have been successfully applied to many practical engineering problems due to their ability to learn by example and their capability of modelling complex functions and processes. This type of network is able to effectively address the curse of dimensionality problem besetting many of the standard modelling techniques used for engine performance mapping. Feed-forward networks, therefore, are proposed in this study as a viable option for performance mapping of normally aspirated direct injection diesel engines.

- Neural network development exacts a number of specific conditions. The two primary conditions being that the training data characterises the problem (function) being modelled, and that it is adequately sized to both train and test the network.

- There is evidence that use of asymmetric activation functions in the hidden layers results in improved learning. Therefore, hyperbolic tangent activation functions are recommended for use. Also, although not conditional when sigmoid activation functions are used, trainable bias terms should be used on the processing elements since there is evidence that network training is improved with their use (specifically on the output processing elements).

- There are many network architectures that can be used for function approximation. However with the engine performance mapping problem at hand and proof of successful prior application, the feedforward backpropagation architecture (multilayer perceptron) was deemed to be appropriate for this modelling problem.

Chapter 4

Neural Network Design Methodology

4.1 Introduction

A multilayer feed-forward neural network trained with the backpropagation algorithm may be viewed as a practical vehicle for performing a *nonlinear input-output mapping* of a general nature (Haykin, 1994). In principle, and under certain practically satisfiable assumptions, feed-forward neural networks with a single hidden layer and trained using the backpropagation algorithm can compute any computable function and will exhibit tolerance of some imprecision (White, 1992; Valiant, 1988; Siegelmann and Sontag, 1999; Orponen, 2000; Sima and Orponen, 2001; Hornick, 1989). The primary assumptions are the availability of appropriate sample data, sample noise (specifically on the output target), number of hidden units, number of weights, and the form of hidden unit activation functions. Unfortunately, there are no theorems that can explicitly answer the question: when and how do neural networks capture problems? Also, there is no theory to tell one how many hidden processing elements are required to approximate any given function. However, there are numerous heuristics, development tools and methodologies available that can be applied in the development of a neural network. In this chapter, these are reviewed and the critical success factors are investigated and discussed in terms of the engine performance mapping problem under investigation.

The development of a neural network usually proceeds by an iterative process of design refinement. There are two reasons for this (Tarrasenko, 1998):

- the full scope of the complexity of the problem may not immediately be apparent;
- it is difficult to predict the performance of a neural network and the way it could vary when changing design (or programming) parameters.

As a result of these, some network design parameters can only be resolved by experimental optimisation. This is because there are several training and topology

parameters that strongly interact with each other and influence whether a feed-forward neural network will converge to a solution or not. With respect to these parameters, and specifically the development of a neural network for diesel engine performance prediction, the neural network's training can be characterised by five parameters:

- Training and validation data sets
- Weight update rate (learning rate)
- Learning algorithm
- Weight initialisation
- Training fitness

and the network's topology (architecture) can be defined in terms of six topology parameters:

- Number of inputs
- Number of hidden layers
- Connections between the layers (fully or partially connected)
- Number of processing elements per hidden layer (complexity)
- Type of processing unit or activation function in each layer
- Number of outputs

The adequate setting or determination of the training and topology parameters can be regarded as the programming phase in the development of a neural network.

When developing a neural network the following issues need to be considered with care (Salustowicz, 1995):

- data requirements and pre-processing;
- data input/output encoding;
- selection of Neural Network type, architecture and learning parameters;
- training the prototype;
- testing the prototype.

These aspects are considered and individually reviewed in the subsections that follow.

4.2 Data Requirements

Data are the lifeblood of any neural network model development. The nature of the data requirements for a neural network application is hugely dependent on the application. In general, both relevant and possibly relevant data should be considered as inputs to the neural network. It is not necessary to know the exact nature of the input-output mapping, only that there is a strong possibility of a relationship. However, one common belief is that, since neural networks are capable of learning, they will be able to determine the input-output mapping (model) of the system or problem domain, irrespective of the number and relevance of input variables to the neural network (Soulié, 1994). This is a common mistake made by non-domain expert users of neural network shells, since the old saying "nonsense in, nonsense out" is as applicable to neural networks as it is for any knowledge acquisition system. Further, it is important to understand that there are no methods for training neural networks that can magically create information that is not contained in the training data. Therefore, designers of neural network models must spend a significant amount of time performing the task of knowledge acquisition.

Selection of input variables is an important and complex task in neural network design. Many neural network designers have indicated a requirement for the extensive use of domain experts for specifying the input variables for the specific application (see Walczak, 1999 for a review). The primary purpose of domain expertise is to guarantee (or reasonably guarantee) that the input variable set is not under or over specified and to avoid noise in the learning process. The benefit of correct specification of the input variables is a network that has improved learning characteristics. In this respect, diesel engine performance expertise has been applied for the selection of the inputs for this study.

4.2.1 Training Data Set Size Requirements

Since neural networks learn from data, the size (number of examples) of the training set directly influences the performance of any model trained non-parametrically, such as a neural network. Typically a neural network requires a lot of data for appropriate training, because there are limited *a priori* assumptions about the data. It is important to know how the size of the training set scales as a function of the size of the network for a given precision in the mapping of the input-output relationship. The

size of the training set has fundamental influence on the practical usefulness of the network. If the training patterns do not convey all the characteristics of the problem, the mapping discovered during training only applies to the training set. Thus the performance in the test set will be much worse than the training set performance. The only general rules that can be formulated are to use a lot of data and use representative data. If one has insufficient data to train the neural network, then the neural network method is not recommended for solving the problem.

An important issue related to proper training is the relationship between training set size and number of weights in the neural network. If the number of training examples is smaller than the number of weights, one can expect that the network may *hard code* the solution, that is, it may allocate one weight to each training example. Although this could lead to good performance on the training data set, it will result in poor performance on a test data set that was not part of the training set. If this occurs, it can be said that the network has poor generalisation capability. Achieving good generalisation can be a complex issue. However, if the training of a neural network can be viewed as a *curve-fitting* problem (Tarassenko, 1998; Haykin, 1994), then such a viewpoint permits the interpretation of generalization as the effect of good nonlinear interpolation of the input data (Wieland, 1987).

4.2.1.1 Generalisation

A network is said to generalise well when the input-output relationship calculated by the network is correct within a specified error for test data never used for training the network (Haykin, 1994). When there is a big discrepancy between the performance in the training set and test set, one can suspect deficient learning or poor generalization. Note that one can always expect a drop in performance from the training set to the test set; however, a large drop in performance of more than 10~15% indicates deficient learning (Haykin, 1994).

Generalisation is influenced by three factors: the size and efficiency (applicability) of the training data set, the architecture of the network, and the physical complexity of the problem at hand (Hush, 1993). In the context of the first two factors, generalisation can be viewed from the perspective of a fixed network architecture (with the number of processing elements optimised) and resolution of determining the size of the training set needed for good generalisation. However, this perspective

ignores the typical problem of data scarcity and data acquisition cost in engineering applications. The perspective taken in this study is one of: the training data set size is fixed (although efficient by using *domain expertise*) and experimental optimisation is used to resolve the number of processing elements (or weight connections) for good generalisation. Resolution of the optimum number of weight connections is not simple and is discussed in further detail in section 4.4. It is appropriate to deal firstly with the heuristics regarding training set size and data partitioning for valid generalisation.

4.2.1.2 Training Set Size Heuristic for Valid Generalisation

Baum (1989) has determined that the appropriate number of training examples is, as a first approximation, directly proportional to the number of weights in the network and inversely proportional to the specified or acceptable error. In this regard, Haykin (1994) has stated in practice all that is needed for good generalisation is to approximately satisfy the condition:

$$N_{\text{train}} \geq \frac{S_n}{\varepsilon} \tag{4.1}$$

where N_{train} is the number of training exemplars (or vectors) required for generalisation, given the number of weights in the network S_n and an input-output mapping error of ε. The number of weights of the network, for an $I-J-K$ multilayer feed-forward network (or MLP) is given by (Tarassenko, 1998):

$$S_n = (I+1)J + (J+1)K \tag{4.2}$$

where I is the number of inputs, J the number of hidden units and K the number of output units in the network. Equation (4.2) shows that the number of required training exemplars increases linearly with the number of free parameters (weight connections) of the network. Information theory suggests that the number of input vectors in the training set should be of the same order as the number of free parameters in the network (Tarassenko, 1998). However, a more conservative rule of thumb states that $N_{\text{train}} \cong 10 \cdot S_n$. That is, the training set size should be 10 times larger than the number of network free parameters to accurately perform on the test

data with 90 percent accuracy (Haykin,1994) or $N_{\text{train}} \cong 20 \cdot S_n$ for a 95 percent accuracy.

The number of training examples will have to be estimated in the first instance since the specification of this number requires knowledge of the final network architecture, which is yet to be created. This is a problem that can be solved by experimental optimisation (trial and error). However, with reference to equation(4.2), the practical limiting quantity is the number of training examples. So most of the time one needs to compromise the number of free parameters of the network to achieve good generalisation performance. That is, the design criteria is to optimise the network size based on the available training set size.

Using the heuristic as defined by equation (4.1) it is assumed that the training set data is representative of all the conditions that could be encountered in the test set. The main focus when creating the training set should be to collect data that covers the full known operating conditions (input space or input region) of the problem one wants to model. If the training set does not contain data from some areas of input space, the neural network performance in those areas will be based on extrapolation. This may or may not correspond to the true output (the desired output). Thus, one should always choose samples for the training set that cover as much of the input space as possible (Haykin, 1994). As with any curve-fitting technique, neural networks cannot extrapolate reliably. If there is no training data in the region from which the test data is taken then, there cannot be any valid generalisation of the test data input (Tarassenko, 1998).

Obtaining a representative data set for good generalisation capability is only the first step. Critical to the development and testing of the neural network is how this data is effectively partitioned into training, validation and test data sets.

4.2.2 Partitioning of Training, Validation and Testing Data Sets

The selection of appropriate training and test data sets is highly important for obtaining valid results. Flexer (1995) states that the minimum guidelines of proper neural network experimentation can be divided into: how to select training and test data, and how to statistically evaluate such experiments, where the former is a

prerequisite for the latter. The data set selection is detailed here and the minimum statistical requirements for neural network evaluation are detailed in section 4.7.

In order to check the generalization capability of a network, a second data set needs to be applied. It is necessary to use different sets of data for training and testing. This is because it is not sufficient to use the re-substitution method where the performance of the trained network is measured on the data set used for training. It is widely known that the performance measure estimated with this re-substitution method is over-optimistic; that is, the same performance measure computed on new, previously unknown data is very likely to produce worse results (Michie, 1994).

Since it is necessary to tune some parameters (learning rate, number of layers, number of processing units) to get the best network performance, a division of the available data into three different sets is recommended (Michie, 1994). The data therefore need to be partitioned into the training, validation and test data sets. The data sets are defined by Tarassenko (1998) as:

- **Training set:** a set of data which is used to train the neural network, that is, for adapting the weights of the network until the stopping criterion is met;
- **Cross-Validation set:** a set of data used to test the performance of the network during training *but not used for modifying the weights of the network*;
- **Test set:** a set of data, which has not previously been applied to the neural network and which is used to test performance. The network performance on the test data measures how well the neural network has learned to generalise.

When preparing data for neural network development, the issue to be resolved is how to effectively partition the available data into these data sets.

4.2.2.1 Data Partitioning

In this research there were a total of eight engine maps comprising the total data set. This data could be allocated randomly between the training, validation and test sets in the ratio 1:1:1 (Tarassenko, 1998) or 40:40:20 (Michie, 1994). However, this approach to partitioning should be applied with care. The problem here is that the engine performance data from all eight engines would appear in all three partitioned

data sets. Strictly speaking, no parts of the test engine data set should appear in the training set or they would form part of the training of the neural network model. If the use of a third independent data set is omitted, the obtained test results will be overoptimistic because the test set used for repeated tuning of the neural network, in fact, becomes a training set. Therefore the generalisation capability of the neural network will not be tested. This problem is solved by holding out the entire data set of one engine for final testing. This is done prior to the random allocation of the data and partitioning. This threefold partitioning of the data is called *double cross-validation* (Michie, 1994). In this study, the generalisation capability of the network-based models is further validated on an alternative test data set that has been generated from the test data set using Delaunay triangulation, so as to generate interpolated target values across the performance map.

Correct selection of the data and its partitioning then supports the goal of the research. That is, can a neural network model generalise a diesel engine performance map when based on key features (or parameters) of a previously unseen engine? Yes, should there be acceptable neural network performance on the test data set. The measures constituting reasonable performance and model validation are detailed and discussed in Chapter 5.

Thus far, some of the requirements for generalisation and data partitioning have been discussed. How the actual data was selected and partitioned is discussed in the next section.

4.3 Diesel Engine Performance Map Data

In this section, the data used for the model development is described. Also, the required transformations of the diesel engine performance map data into the input variables are defined together with the data partioning.

4.3.1 Actual Engine Data and Partitioning

Eight normally aspirated direct injection diesel engines of different rated power and cylinder number were chosen for the modelling and testing processes. Six of these engines are manufactured by KHD Deutz and two are manufactured by MAN. The relevant geometric and performance parameters for each engine are given in Table A.1.1 of Appendix A.1. The performance maps corresponding to the engine data of Table A.1.1 are shown in Appendix A.2 (Figures A.2.1 to A.2.8).

The engine data for the model development were obtained by digitizing the engine maps as shown in Appendix A.2 and then applying a digital terrain modelling (DTM) process to generate the correct coordinates for the constant brake specific fuel consumption lines (or contour lines). These define the engine fuel consumption performance on the two-dimensional map. Software for the DTM process was supplied by IntelliCAD® 2001 (CADopia).

The first issue with the data partitioning relates to the selection of an engine for the test data set. The reason for having a completely independent data set has been explained in section 4.2.2.1 as per the *double cross-validation* method of Michie (1994). Therefore via a process of random selection from the population of eight engines, engine number 4 was selected for the test data set. Therefore, using Table A.1.1, engine numbers 1, 2, 3, 5, 6, 7, and 8 were used for training the neural network and engine number 4 was used to test the generalization capability of the trained network. Engine number 4 has further been used to compare the neural network model performance to that estimated using the quadratic method as proposed by Golverk (1994a).

The second issue with the data partitioning relates to the split between the training and validation data. Using a 1:1 (or 50%:50%) split between the training and

validation data sets is advocated (Tarassenko, 1998; Michie, 1994). However, in the context of this research this is wasteful of the available data. Since the neural network size is constrained by the training data set size (see section 4.2.1.2), and since the network is required to learn the characteristic of the diesel engines, an 85%:15% split of the data has been made. Therefore there is over five times more data allocated to training than to validation. This allocation is justified, since an independent test data set is available.

The inputs and their corresponding output, the exemplar pairs, were randomised and listed in this random order. An approximate 85%:15% cut-off was applied to the randomly ordered list so as to define the training and validation data sets. The results of the data partitioning are summarised in Table 4.1.

Table 4.1: Partitioned Data Sets

Data Set	**Size (exemplars)**	**% of Total Data**
Training	8 855	80
Validation	1 563	14
Testing (independent)*	699	6
Total	**11 117**	**100**

** The double cross-validation data set*

4.3.2 Modelling Data – Engine Parameters and Model Input Variables

The engine map data have to be transformed corresponding to the required neural network model input variables. These variables and the corresponding transformations are shown in Table 4.2.

Table 4.2: Neural Network Model Input and Target Variables

Variable Name	Variable Description	Units	Variable Range (Training Data)	Equation Reference	Table A.1.1 Column Reference
x_1	Mean Piston Speed	$m.s^{-1}$	3.3 – 13.8	(2.14)	
x_2	Break Mean Effective Pressure	Bar ($100\ kN.m^{-2}$)	107 - 850	(2.11)	
x_3	Specific Power	$kW.m^{-2}$	93 - 2194	(2.20)	
x_4	Piston area to volume displaced	m^{-1}	7.7 – 12.5	(2.24)	11
x_5	Standard Rated Specific Power	$kW.m^{-2}$	1706 – 1882	(2.21)	13
x_6	Standard Rated Specific Fuel Consumption	g/kW.h	246 – 288	(2.23)	16
d (target)	Brake Specific Fuel Consumption	g/kW.h	211 - 400	(2.18)	

The training input vector is therefore defined as:

$$\mathbf{x}(n) = \left[x_1(n)\ x_2(n)\ x_3(n)\ x_4(n)\ x_5(n)\ x_6(n) \right]^{\mathrm{T}} \tag{4.3}$$

where n is the input exemplar number.

For continuous variables, as is the case here, it is usually sufficient to apply the variable directly to the input of the neural network. However, the ranges of the input variables differ (as shown in Table 4.2). This suggests that a suitable normalisation or transformation should be applied so that the transformed variables all cover the same range. This transformation is called variable pre-processing.

4.3.2.1 Variable Pre-processing

Neural networks very rarely operate directly on the raw input data. There is usually an initial pre-processing procedure applied to the raw data such as normalisation or standardisation.

4.3.2.1.1 Input Data Normalisation or Standardisation

If the input variables are combined linearly, as in an MLP, then it is strictly unnecessary to standardise the inputs, at least in theory. The reason is that any rescaling of an input vector can be effectively undone by changing the corresponding network weights and biases, leaving the activation function with the exact same outputs as before. This is because, in principle, the weights should be able to adapt to the different dynamic ranges of the input. However, there are a variety of practical reasons why standardising the inputs can make training faster and reduce the chances of getting stuck in local minima of the surface defined by the cost function.

The contribution of an input will depend heavily on the scale of its variability relative to other inputs. From Table 4.2 it is seen that the relative ranges of the inputs are very different. For example, the contribution of the specific power input to the calculated distance, or the inner product of equation (3.5), would be significantly greater than that of the mean piston speed input. So it is essential to rescale the inputs so that their variability reflects their importance, or at least they are not in inverse relation to their importance, as would be the case if the two inputs mentioned were not standardised. For lack of better prior information, it is common to standardise each input to the same range or the same standard deviation.

Another problem is processing element saturation. When sigmoid functions are used, the weighted summation of the hidden units may turn out to be very large (positive or negative). Since the derivative of the sigmoid activation function is zero for very positive (or very negative) values of the weighted summation, the summation will remain at large values. Hence the network weights do not update, or update very slowly. This is the problem of *stuck units*, with learning proceeding very slowly or not at all (Tarassenko, 1998). This problem can be addressed by ensuring that the weights are initialised with small random values. These random values depend on the scale of the inputs and the number of inputs as well as their correlations. Standardising the inputs removes the problem of scale dependence of the initial

weights (Thimm, 1994). Further, it is important to center the inputs to get good random initialisations. In particular, scaling the inputs to [-1,1] will work better than [0,1], although any scaling that sets to zero the mean or median or other measures of central tendency is likely to be as good. In this regard, robust estimators of location and scale will be even better for input variables with extreme outliers (Iglewicz, 1983).

Here, the transformation applied to the inputs is the zero-mean unit-variance transformation, which is a linear transformation independently applied to each input variable x_i (see Table 4.2 for the input variables). The normalised value of x_i is given by:

$$x_i^* = \frac{x_i - \mu_i}{\sigma_i} \tag{4.4}$$

where i is the variable number, and the mean μ_i and the standard deviation σ_i are calculated for the training set and not from the validation or test data sets. The validation and test data are standardised using the statistics computed from the training data before being applied to the network. Since there is only one target variable, the specific fuel consumption, no standardisation has been applied to the target values. Hence a linear activation function on the output layer has been used in this study.

4.3.2.1.2 Input/output encoding

For a network to be able to generalise correctly, an important criterion is that the training data should be coded in a form which allows for interpolation (Tarassenko, 1998) – see 4.2.1. To accomplish this, the randomised training input-desired exemplars have been ordered by increasing target or desired value, that is, by increasing specific fuel consumption. This is contrary to the heuristic that suggests that the training exemplars constituting the batch should be randomised before each presentation of the batch to the network (Haykin, 1985); the result of which is stochastic learning.

After the careful selection of the input variables (based on the domain expertise approach), partitioning and the pre-processing, the next phase of the model development involves the selection of the neural network type and architecture.

4.4 Selection of NN Type, Architecture and Size

The selection of the neural network type, architecture and size is a critical to the development of a neural network model. This aspect of the neural network development is investigated and discussed in this section.

4.4.1 Selection of NN Architecture (Topology)

A topology well suited to the problem at hand is highly important for the specific neural network to find a solution. Networks that are too small are not able to learn the training data set. On the other hand, networks that are too large (complex) can suffer from poor generalisation after training. Further, networks of equal size may or may not be able to find a solution, depending on their topology. The optimal topology of a network is the one with the smallest complexity which still allows the network to learn the training data and generalize the test data. Such a topology can be hard to find (Salustowicz, 1995). In the context of this research, the interest is in the generalisation performance of the trained network on a test set of previously unseen input data. Therefore, the network architecture must be chosen that will give the best generalisation performance for a given set of input variables.

Cybenko (1989), Funahashi (1989) and Hornik (1989) rigorously stated the universal approximation theorem for multilayer perceptrons (MLP). This theorem, in effect, states that a single hidden layer with sigmoid non-linearity is sufficient to compute a uniform error approximation to a given training set represented by a set of inputs and a desired output. That is, it can approximate any function with arbitrary accuracy. This theorem is an existence theorem (with the number of hidden units being unconstrained), so it does not address the engineering question: how many processing elements and layers does the MLP need to solve the problem? This remains an open question for which experimentation is necessary. However, it is extremely important to know that, theoretically a MLP is a universal approximator, and that one hidden layer is all it takes to reach this arbitrary input-output mapping capability.

The problem with MLP's using one hidden layer is that the processing elements tend to interact with each other globally. In complex problems, this interaction makes it difficult to improve the approximation at one point without worsening it at some other point. However, Chester (1990), Sontag (1992) and Funahashi (1989) have demonstrated and justified that an MLP with two hidden layers can often yield an accurate approximation with fewer weights than an MLP with one hidden layer, even for inverse problems requiring discontinuous functions. The hidden layers tend to perform as follows:

First hidden layer: here, *local features* are extracted – some processing elements partition the input space into regions and others learn the local features defining these regions.
Second hidden layer: here, *global features* are extracted – a processing element combines the outputs of the first layer operating on a particular region of the input space, and thereby learns the global features for that region.

Basically, the layers act as feature detectors of the problem or function being approximated.

In light of this, this research problem could therefore reduce to defining a two layer MLP and choosing the number of hidden processing elements in the layers. This is then an issue of model order selection and will depend on the complexity of the underlying function which the neural network is approximating (which is unknown). Unfortunately, using two hidden layers exacerbates the problem associated with local minima, and it thus becomes important to use lots of random weight initialisations or other methods for global optimization. These problems are addressed in this research by using multiple training run (multiple network initialisation) and Genetic Algorithm approaches – a discussion on which is deferred until section 4.6.2.

Several neural network simulation studies show that many problems cannot be solved by a learning algorithm in conventional fully connected layered neural networks (Dodd, 1990 and Solla, 1989). Two problems that occur frequently are: a lack of generalisation and a problem called *interference*. These problems are addressed by considering modularity in networks.

4.4.2 Modular Networks

The class of feedforward networks can be divided into two sub-classes; modular and non-modular feedforward networks. An example for each of these two sub-classes is presented in Figure 4.1 (Salustowicz, 1995). The network in figure 4.1 (a) is a non-modular network, as it contains no sub-networks. The adjacent layers of this network are completely connected to each other. The network in figure 4.1 (b), on the other hand, is a modular network, as it contains two independent sub-structures (A and B), which are not connected to each other.

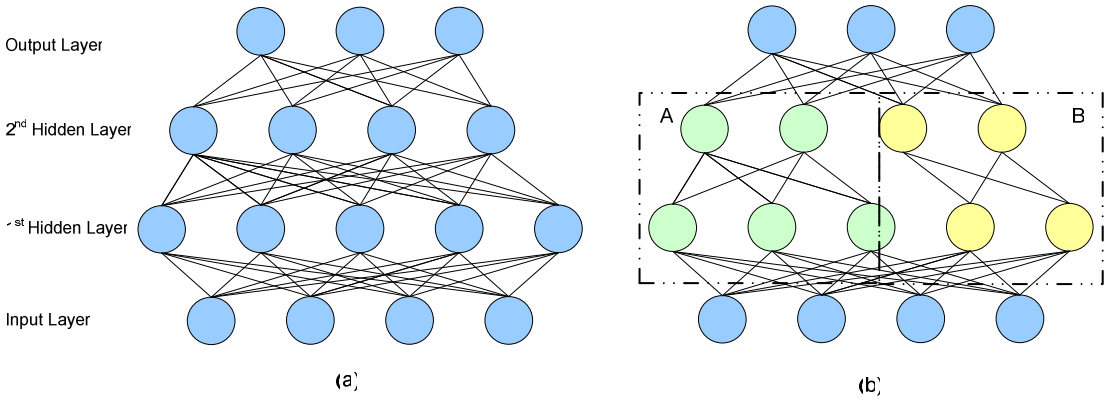


Figure 4.1: Non-Modular and Modular Networks (Salustowicz, 1995)

Modularity is therefore used here in the context of dividing networks into sub-networks that are each expected to solve parts of the problem by itself.

4.4.2.1 Advantages of Modular Networks

Modular feed-forward networks have some advantages over their non-modular counterparts and a few of those advantages are as follows (Boers, 1992):

4.4.2.1.1 Lower Complexity

One major advantage of modularity is lower complexity (the complexity of a network is defined here by the number of processing elements and not by their arrangement in the topology). If a modular network is able to capture a problem at all, it has the advantage that due to its lower complexity, the solutions search space is reduced (the search space of solutions is defined by all combinations of weight and bias values). The fewer weights and biases a network has the smaller is the search space, resulting in fewer local minima. The desired solution can therefore be found

faster and more reliably when training with a local gradient-descent learning algorithm like backpropagation. A large enough network will always learn the training data, but it may not find the desired solution. It is more likely to find a solution which will fit the training data but perform poorly on the generalisation task. Furthermore, networks of smaller complexity are less likely to be affected by the over-learning effect. Over-learning a problem means that after training, the training data are approximated too well by the network, and correspondingly the network performs poorly on the generalisation task. Basically, modular networks train better due to lower complexity.

4.4.2.1.2 Interference Avoidance (Spatial and Temporal Crosstalk)

Another advantage is the capability of modular networks to cope with interference or crosstalk problems. Jacobs (1990) distinguished two types of crosstalk: spatial and temporal. Spatial crosstalk occurs when the output units of a network provide conflicting error information to a hidden unit. Temporal crosstalk occurs when units receive inconsistent training information at different times.

Interference problems can occur when a non-modular network is trained to learn two independent parts of a problem (Rueckl, 1989). A network, which is originally large enough to capture the problem, cannot be trained as it *forgets* one part of the problem while learning the other one. The network shows an oscillating behaviour, as it is not able to capture both problems at once. Modular networks often constitute a good solution to interference problems as they can assign one sub-network to learn one part of the problem and another sub-network to learn the other part.

In this research, the only way of obtaining reliable data is by taking data from the specific fuel consumption contours (constant specific fuel consumption lines) of the engine performance maps. Hence, spatial and temporal interference problems were anticipated due to the construct of the available engine performance data, where different inputs (different engine parameters) map onto the same output (the specific fuel consumption at a contour). Therefore, due to the apparent benefits of modular feedforward networks, both a non-modular and a modular network have been developed so as to enable assessment of any apparent benefits of modular networks in the application of this study.

4.5 The Neural Network Models

Figure 4.2 shows a block diagram of a standard fully connected two hidden layer feedforward network. It has hyperbolic tangent activation functions in the hidden layers and a pure linear output activation function. This is the non-modular network architecture (or topology) used in this study.

Figure 4.3 shows a block diagram of a modular network consisting of two fully connected two hidden layer feedforward networks. It has hyperbolic tangent activation functions and linear output activation function. There is also a weighted connection directly from the input layer to the output layer. This is the modular network topology used in this study.

There are many ways to segment a modular network. However, it is unclear how to best design the modular topology based on the data. Also, there are no guarantees that each segmented module is specialising its training on a unique portion of the data (Boers, 1992). Therefore, without any basis to exclude symmetry in the topology, a symmetrical modular network topology has been defined. The direct connection between the input layer and linear output activation function enables three calibrating parameters to more directly effect the output without interfering with the operation of the other modules. The three engine performance calibrating parameters for the model are: x_4, x_5 and x_6 of Table 4.2.

Note that only the layouts or topology of the networks shown in Figures 4.2 and 4.3 have been defined. The optimal number of processing elements and weights/biases for good learning and generalisation remain to be established.

The property of primary significance in using neural networks as models is the ability of the network to learn from its environment and improve its performance through learning over time using some prescribed method. In the next section, this learning property of neural networks, and how to best train them, is presented and discussed.

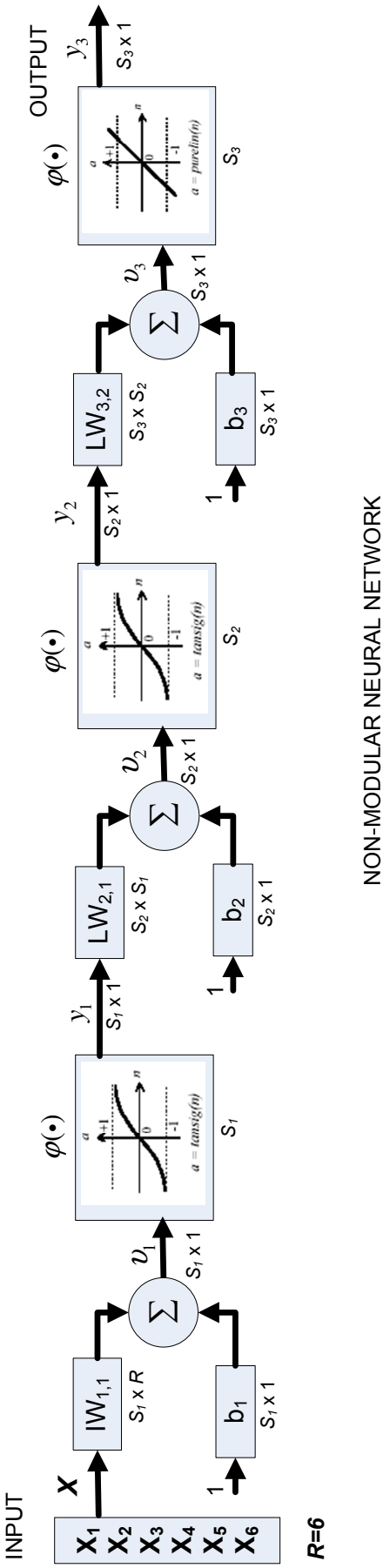


Figure 4.2: Non-Modular Feed Forward Neural Network Model

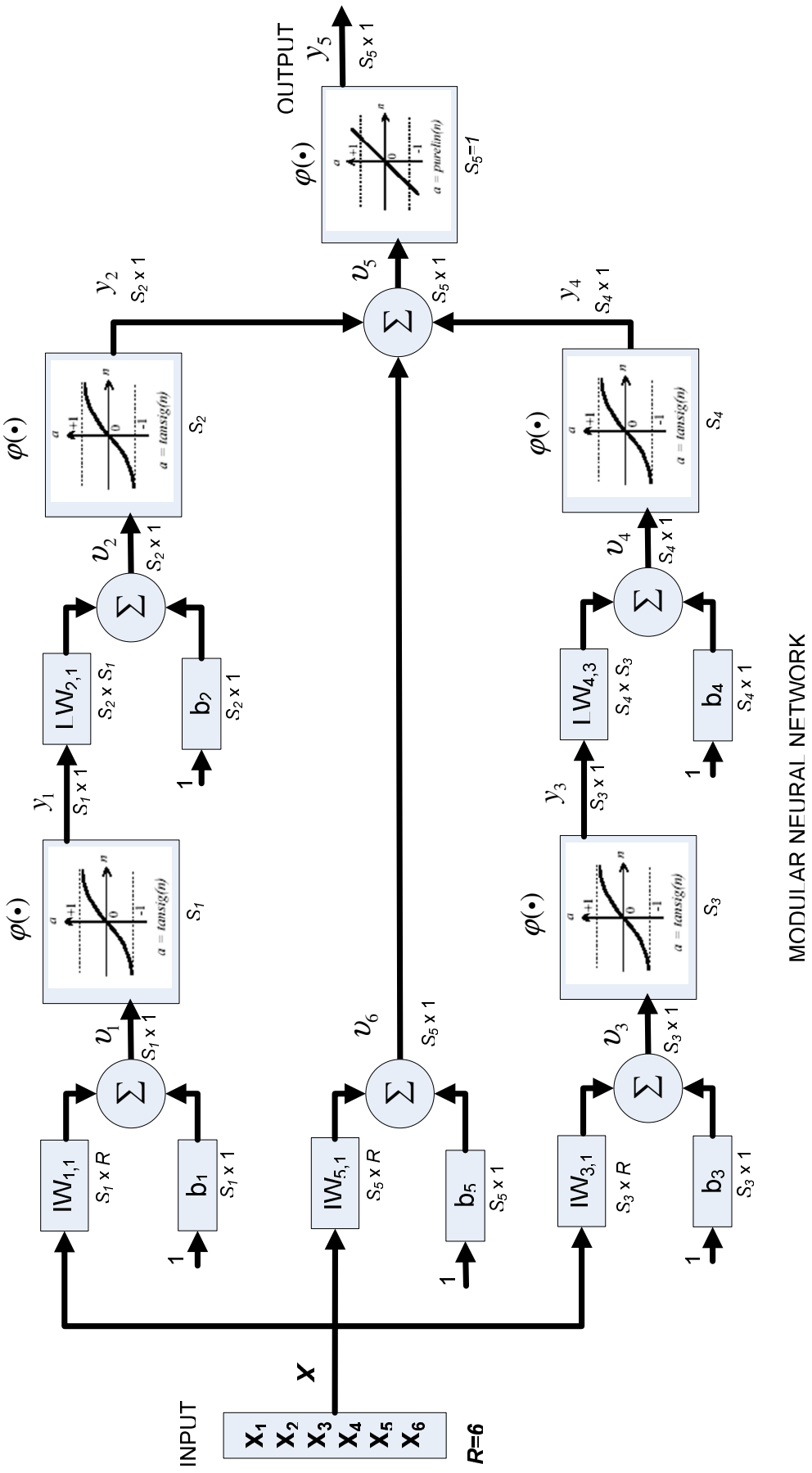


Figure 4.3: Modular Feed Forward Neural Network Model

4.6 Network Learning (Training) and Optimisation

In engineering applications, where the desired performance is known, supervised learning becomes very important. Here the multilayer perceptron (feedforward networks) applied to the diesel engine performance modelling problem have been trained in a supervised manner with a highly popular algorithm known as the *error back-propagation algorithm* based on the *error-correction learning rule*. The derivation of the algorithm is beyond the scope of this work and can be found in numerous founding papers and textbooks on the subject, for example, Werbos (1974), Rumelhart, Hinton and Williams (1986), LeCun (1985) and Haykin (1994).

The beauty of training using the backpropagation algorithm is that it is a systematic procedure that can be applied independent of the topology of the network and the input dimensionality. Larger, more complex networks may require more training time, but backpropagation is still fully capable of training them. Although backpropagation is popular and has successfully solved difficult problems, unfortunately it is slow for practical applications and may never capture the problem at hand (Schiffmann, 1993). Therefore, in the context of this study, it is pertinent to investigate learning in further detail and establish means of improving the neural network's ability to capture (learn) the engine performance mapping problem.

Haykin (1994) adapted a definition of learning from Mendel and McClaren (1970) as follows:

> *Learning is a process by which the free parameters of a neural network are adapted through a continuous process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place.*

This definition points towards learning requiring three ingredients. First, as the network parameters change, the performance should improve (error reduction). Therefore, the definition of a measure of error is required. Second, the rules for changing the parameters need to be specified. And finally, the procedure for training the network should be done with known data.

Dealing with the issue of defining the measure of error performance: when the performance function is based on the definition of an error measure, learning is said to be supervised. Normally the error is defined as the difference between the output of the neural network and a pre-specified external desired value (target). In this research and specific to the modelling problem at hand, the learning process used consisted of error correction learning (the learning rule) and supervised learning (the learning paradigm); and being specific:

Let $d_k(n)$ denote the desired or target response for processing element k at epoch n (an epoch is defined as the presentation of all the training exemplars at the input of the network in batch-mode of backpropagation learning). Further, let $y_k(n)$ be the corresponding actual response of this neuron to an input vector $\mathbf{x}(n)$. Hence one may define an error signal as:

$$e_k(n) = d_k(n) - y_k(n) \tag{4.5}$$

The primary purpose of error correction learning is to minimise (optimise) a cost function based on the error $e_k(n)$. A practical approach to this optimisation is to settle for the *average squared error* cost function defined as (batch-mode)

$$\xi_{\text{AV}} = \frac{1}{2N}\sum_{n=1}^{N}\sum_{k\in C} e_k^2(n) \tag{4.6}$$

where C is the number of processing elements in the output layer of the network. Therefore, the error reported to the supervised learning procedure is simply the sum of the squared Euclidean distance between the network's output and the desired response. The network is then optimised by minimising ξ_{AV} with respect to the connection (or synaptic) weights.

Dealing with the issue of weight change; and being specific: with reference to Figure 3.2, let $w_{kj}(n)$ denote the synaptic weight w_{kj} at epoch n. Thus at epoch n, an adjustment $\Delta w_{kj}(n)$ is applied to the synaptic weight $w_{kj}(n)$ connecting processing element j to processing element k, thereby producing an updated weight value $w_{kj}(n+1)$, or:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{4.7}$$

where $w_{kj}(n)$ and $w_{kj}(n+1)$ can be considered as the old and new values of the synaptic weight w_{kj} respectively.

For a learning rate parameter η, the adjustment applied to the synaptic weight w_{kj} is defined by the *Delta Rule*

$$\Delta w_{kj} = -\eta \frac{\partial \xi_{\mathrm{AV}}}{\partial w_{kj}} = -\frac{\eta}{N} \sum_{n=1}^{N} e_k(n) \frac{\partial e_k(n)}{\partial w_{kj}} \tag{4.8}$$

That is, the backpropagation is gradient decent learning where η is a positive constant that determines the rate of learning.

Since error-correction learning behaves like a closed feedback system, stability of the process can only be assured by careful choice of the value given to the learning rate parameter η. In fact, the learning rate parameter has a profound impact on the performance of error-correction learning in that it affects both the rate of convergence of learning as well as the convergence itself (Haykin, 1991).

The performance of backpropagation is significantly affected by the presence of *local minima* (isolated valleys) as well as global minima on the geometric error surface defined by the error function. Since backpropagation is a steepest decent search algorithm, it runs the risk of being trapped in a local minimum and thereby fails to converge. It is clearly undesirable for the learning process to terminate in a local minimum. Auer (1996) has shown that the number of local minima may grow exponentially in the number of parameters (weights and biases).

Therefore it may be advantageous, in terms of network performance, to employ alternative learning algorithms to the problem with the focus on effectively facilitating the finding of the error global minimum.

4.6.1 Network Weights Adaptation (Learning Algorithm)

The weight adjustment and update can be accomplished via learning algorithms, and for a specified neural network there is no unique learning algorithm. There are a diverse range of learning algorithms available that differ in the way that the adjustment (Δw_{kj}) to the synaptic weight (w_{kj}) is calculated.

Dealing with the issue of the rules on how learning parameters are updated, there are four extremely useful heuristics due to which, if applied, can accelerate convergence through learning rate adaptation (Jacobs, 1988):

Heuristic 1. Every adjustable network parameter of the cost function should have its own individual learning rate parameter.
Heuristic 2. Every learning rate parameter should be allowed to vary from one iteration (epoch) to the next.
Heuristic 3. When the derivative of the cost function with respect to the synaptic weight has the same algebraic sign for several consecutive iterations of the algorithm, the learning rate parameter for that particular weight should be increased.
Heuristic 4. When the algebraic sign of the derivative of the cost function with respect to a particular synaptic weight alternates for several consecutive iterations of the algorithm, the learning rate parameter should be decreased.

There exist several possible implementations of these heuristics. One of which, called the delta-bar-delta learning rule of Jacobs (1988), is informally presented here. This locally adaptive algorithm consists of both a weight update rule and a learning rate update rule.

The weight update rule:
At epoch n, let $\xi(n)$ denote the cost function defining the error surface, $w_{kj}(n)$ denote the value of the synaptic weight connecting processing element j to k, and $\eta_{kj}(n)$ the learning rate parameter corresponding to $w_{kj}(n)$. The the delta-bar delta update rule can then be written as

$$w_{kj}(n+1) = w_{kj}(n) - \eta_{kj}(n+1)\frac{\partial \xi(n)}{\partial w_{kj}} = w_{kj}(n) - \eta_{kj}(n+1)\nabla w_{kj}(n) \qquad (4.9)$$

The learning rate update rule:

At epoch n, let $w_{kj}(n)$ denote the value of the synaptic weight connecting processing element j to k. Let $\eta_{kj}(n)$ denote the learning rate parameter assigned to this weight at this iteration. The delta-bar-delta learning rule is then defined as:

$$\Delta\eta_{kj}(n) = \begin{cases} \kappa & \text{if} \quad S_{kj}(n-1)\nabla w_{kj}(n) > 0 \\ -\beta\eta_{kj}(n) & \text{if} \quad S_{kj}(n-1)\nabla w_{kj}(n) < 0 \\ 0 & \text{if} \quad \text{otherwise} \end{cases} \tag{4.10}$$

where:

$$S_{kj}(n) = (1-\zeta)\nabla w_{kj}(n-1) + \zeta S_{kj}(n-1) \tag{4.11}$$

and:

$$\kappa - \text{additive constant}$$
$$\beta - \text{multiplicative constant}$$
$$\zeta - \text{smoothing factor}$$
$$(n-1) - \text{previous iteration}$$

In the learning rate update rule of equation (4.10), $\nabla w_{kj}(n)$ is the partial derivative of the error with respect to $w_{kj}(n)$, and $S_{kj}(n)$ is an exponential average of the current and past derivatives with ζ as the base and n as the exponent.

The learning rate and weight update are adapted according to the previous values of the error at the processing element. If the current and past weight updates are both of the same sign, it increases the learning rate linearly. The reasoning is that if the weight is being moved in the same direction to decrease the error, then it will get there faster with a larger step size. If the updates have different signs, this is an indication that the weight has been moved too far. When this happens, the learning rate decreases geometrically (exponentially) to avoid divergence.

Schiffmann (1993) demonstrated on a difficult problem, that a group of locally adaptive algorithms (of which delta-bar-delta is one such algorithm) had superior performance over standard backpropagation using gradient decent. Due to the potential benefits as defined by Haykin (1992), Jacobs (1988) and Schiffmann (1993), the delta-bar-delta learning rule has been adopted for network training in this study.

Although an effective training algorithm has been selected for use, it can only be put to work on a predetermined network topology. As previously discussed, network size has a profound effect on whether a network can capture a problem and generalise well on previously unseen data. Therefore, selecting an appropriate training algorithm and using network modularity are only part of the required solution. This is because, in this engine performance mapping problem, the network size is constrained by the training set size. The problem that remains to be resolved is: what is the optimum network size given the problem at hand and the available training data?

If the problem of finding an optimal network size can be thought of as a complex search problem (where the search space is the space of all possible network sizes, and where the goal is to minimize an error function while preserving generalization capabilities), then there are two ways to approach this complex search problem:

> ***Apply experimental optimisation*** – construct numerous networks with differing size and learning parameter values, and via a process of iterative design refinement and performance measurement attempt to optimise the solution. Usually, the neural network connectivity and the activation functions are fixed by the neural network developer and the only parameters for model selection are the number of neurons for each layer and, in some occasions, the number of layers. Those parameters are then optimized with respect to the dataset with a trial and error procedure or using cross–validation (Matteucci, 2002). Or;
>
> ***Apply a search algorithm*** – use a defined search algorithm to find the optimum network size and learning parameter values in order to optimise the solution.

The experimental optimisation approach is a standard method applied to neural network development; however, it is *ad hoc* or based on trial and error and most significantly requires the network developer to be highly expert. Thus, by applying experimental optimisation, neural network development remains an art more than a science. There is nothing fundamentally flawed with this approach and it has had proven results over many years on engineering applications.

Although there are many heuristic and numerical techniques available for optimisation, there are experimenters using the mechanics of natural selection (genetic algorithms) to optimize neural networks and engineering problems. This can be seen from the numerous papers on the topic (Tabakov, 2001; Yao, 1992; Lock and Giraud-Carrier, 1998; and Balakrishnan *et al.* 1995).

Matteucci (2002) highlights that genetic algorithms have been used to optimize almost all the parameters that characterise a neural network; for example, weights, learning and topology. Based on the above foundations, genetic algorithms are investigated in further detail for this research problem.

4.6.2 Genetic Algorithms

Researchers have demonstrated the combination of genetic algorithms with neural networks to form more powerful adaptive systems, for example, Dodd (1990); Macfarlane (1992), Miller (1989); Whitley and Starkweather (1990). Others have taken a more direct approach, where not only the topology and learning parameters are specified by the genetic algorithm, but also all weight values and thresholds, for example, De Garis (1990), Harp (1989), Lock and Giraud-Carrier (1998), Yao (1992, 1999), and Yao and Liu (1997).

Genetic algorithms were introduced by John Holland (1976) and his colleagues, and have found numerous applications in neural network development. Genetic algorithms are defined by Goldberg (1989, p.1) as "... search algorithms based on the mechanics of natural selection and natural genetics". The process of evolution is used as a *real-world model* that serves as a source of ideas for solving practical and theoretical problems in modeling and optimization. Just as neural networks use a brain metaphor, genetic algorithms use an evolution metaphor (Happel, 1994).

Goldberg (1989, p.7) mentions four differences between genetic algorithms and other search methods.

- Genetic algorithms work with a coded parameter set.
- They search from a population of points in a solution space, rather than from a single point.
- They only use directly available information provided through a fitness function.
- They rely on probabilistic transition rules instead of deterministic rules and are therefore not subject to the pathologies typically encountered with learning rules.

Genetic algorithms have proved to be a powerful search tool when the search space is large and multimodal, and when it is not possible to write an analytical form for the error function in such a space. In these applications, genetic algorithms excel because they can simultaneously and thoroughly explore many different parts of a large solution space seeking a suitable solution. At first, completely random solutions are tried and evaluated according to a fitness function, and then the best ones are combined using specific genetic operators. This gives the ability to adequately explore possible solutions while, at the same time, preserving from each solution the parts which work properly. The key processes in the use of genetic algorithms for the evolutionary design of neural networks are shown in Figure 4.4 (Balakrishnan, 1995).

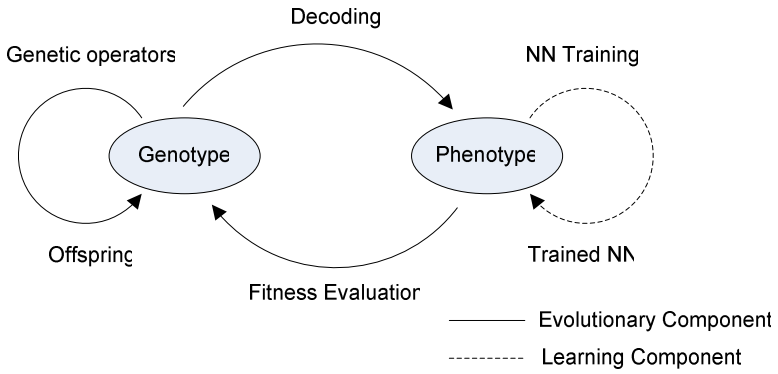


Figure 4.4: Neural Network – Process of Evolutionary Design

In order to solve the problem using genetic algorithms, the phenotype (in this case the parameters defining the neural network and its learning) must be coded onto a genotype (G_n). A genotype is typically represented by a string of binary features and discrete or continuous real numbers. The genetic algorithm works with a population of genotypes ($G_1, G_2, \ldots, G_n$ where n is an even number and each genotype encodes a *possible* solution to a neural network). Using the genotypes, the genetic algorithm searches globally for the optimum solution by applying genetic operators to the genotypes with respect to their fitness (or quality). That is, starting from an initial population that may consist of random genotypes, the search for optimal solutions proceeds through the general process of reproduction. New genotypes are created and replace less fit solutions in the present population from generation to generation (evolution). This leads to a higher overall fitness of the population and to better solutions.

The power of a genetic algorithm is directly related to the choice of the genetic representation (how the parameters of the phenotype are coded onto the genotype). A detailed discussion of genetic coding is beyond the scope of this study; however there are several criteria that result in efficient genetic coding (see Salustowicz, 1995 for further insight to these criteria).

4.6.2.1 Genetic Algorithm Strategy Employed

Learning in neural networks, when viewed as a search method using a learning algorithm, is often sensitive to local minima of the error surface. Also, constraints on network complexity form a major determinant of the learning capacity of a neural network. Therefore, specification of this complexity by a genetic algorithm, seems to be a good way to combine evolutionary learning with neural learning. If complexity is defined as the number of weighted connections between the layers, then neural learning imparts a finer structure on a neural network coarsely outlined by a genetic algorithm. Thus, the genetic algorithm moves the search to an appropriate region in the solution space. Learning then executes a more local search to achieve an optimal performance. Genetic algorithms can be used for network weight optimisation, but they are more suitable for global search instead of local optimisation. It is preferable not to use genetic algorithms for weight optimisation but rather to evaluate the error function and its derivative and effectively apply the commonly accepted learning optimisation heuristics (Matteucci, 2002; Harp, 1989;

Lock and Giraud-Carrier, 1998; Salustowicz, 1995; Walczak, 1999). Therefore, the approach taken for the neural network optimization was as follows:

- Employ a genetic algorithm to adjust the complexity of the network, defined here by the amount of connections (number of weighted connections) and not their arrangement in the topology. That is, the general structure (arrangement) of the neural network was defined (see Figures 4.2 and 4.3) and a genetic algorithm was used to optimise the number of weighted connections in this predefined structure.

- Use the genetic algorithm to select the learning parameters during training.

- Use the delta-bar-delta learning method as defined in section 4.6.1 to optimize the weights. That is, use a local search method to optimise the weight values after their number has been optimised by the genetic algorithm.

Using a genetic algorithm to adjust the learning parameters can be justified as follows:

The control of the learning parameters is an unsolved problem in neural network modelling, and for that matter, in optimization theory. An objective is that the network trains as fast as possible and reaches the best performance. Increasing the learning rate parameter will decrease the training time, but will also increase the possibility of divergence, and of rattling around the optimal value. Since the weight correction is dependent upon the performance surface characteristics and learning rate, to obtain constant learning, an adaptive learning parameter is necessary. Although Jacob's delta-bar-delta is a versatile procedure for this, it requires care in specification of the learning parameters (κ, β, ζ). Thus, the learning rate parameters need to be experimentally set. This will tend to reduce the efficiency of the genetic algorithm employed, since there will be a trade-off between the genetic algorithm weight space search and the experimental learning rate space search. Therefore, the genetic algorithm has been used to optimise the learning rate parameters.

Now, considering the input variables. The genetic algorithm is certainly capable of finding an optimum solution to the number of input variables or finding the most important variables (see Lock and Giraud-Carrier, 1998). Therefore it is valid to apply the genetic algorithm to optimise the input variables. However, in this study, the approach taken was to apply domain specific expertise to define the variables that characterise diesel engine performance (provide *a priori* information to the network) and then use statistical tests (sensitivity analysis and variable perturbation) to check variable validity. This was further supported by careful data partitioning and the application of an independent test data set for network generalisation verification.

Of the thirteen parameters characterising the training and topology of the neural network (see 4.1), the genetic algorithm was used to optimise the following four parameters:

Learning parameters:

- Learning rate increase parameter (κ)
- Learning rate decrease parameter (β)
- Learning rate smoothing parameter (ζ)

Topology parameter:

- Number of processing elements per hidden layer (S_n)

Therefore the coded solution (genotype) takes on the following sequence for the non-modular and modular networks respectively:

$$S_1\ S_2 \quad a_1\ a_2\ a_3$$

$$S_1\ S_2\ S_3\ S_4 \quad a_1\ a_2\ a_3$$

where $S_1\ S_2\ S_3\ S_4$ are the number of processing units in the hidden layers, $a_1\ a_2\ a_3$ are the delta-bar-delta learning rate adaptation terms.

4.6.2.2 Scheme of the Genetic Algorithm

Figure 4.5 shows the scheme for the genetic algorithm search when combined with the neural network training. It is used to support the description of the genetic algorithm scheme of this research problem, and is based on the strategy defined in 4.6.2.1.

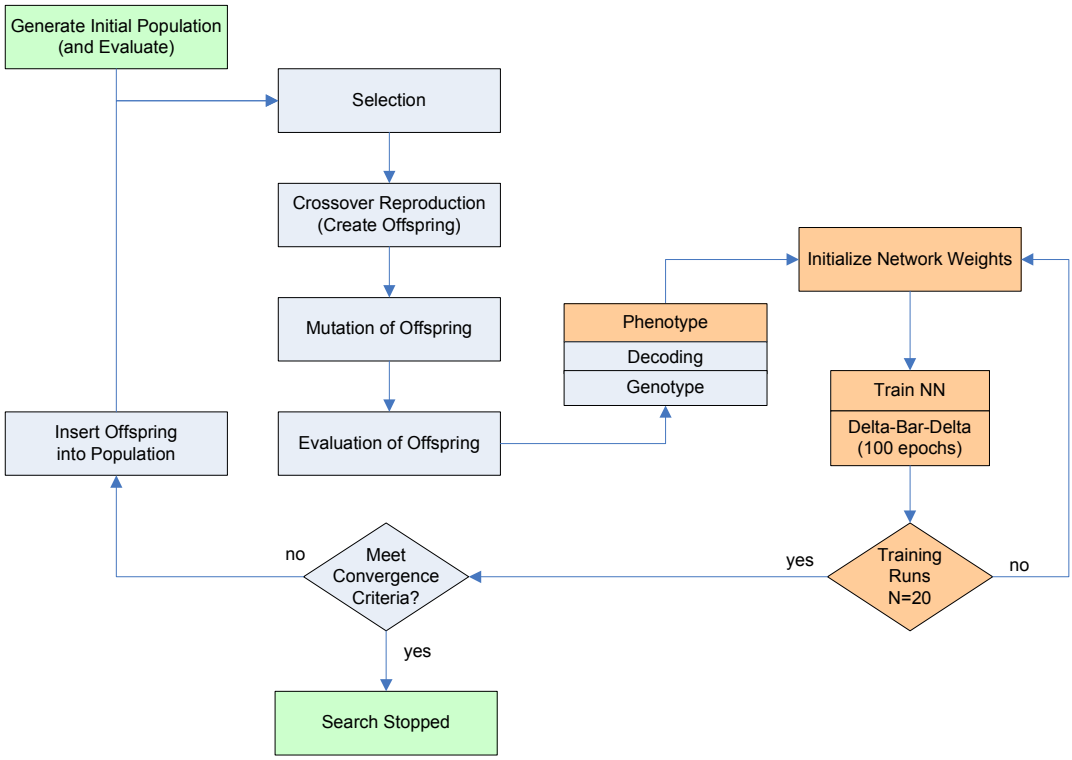


Figure 4.5: Scheme for the Genetic Algorithm Strategy

4.6.2.2.1 Initial Population and Evaluation

The initial population of genotypes (of possible solutions) is generated randomly using a uniform distribution. The number of processing elements in each layer is randomly generated, as are the learning rate parameters for each member of the population using a uniform distribution. Note that random solutions are constrained so that *invalid* solutions are not coded. This addresses the issue of *closure* when using genetic algorithms (every genotype can be decoded into a valid phenotype or solution, thereby keeping the search space to a minimum). Since the network size has been constrained by the size of the training data set size, the number of

processing elements in each layer has been limited via an estimation of the maximum allowable processing elements using the heuristic defined by equation (4.1) - see section 4.2.1.2. Further, the learning rate parameters of the delta-bar-delta learning rate adaptation algorithm have been constrained to fall within the heuristics of optimal learning (for example, the learning rate parameters are not allowed to be zero, since this will result in simple backpropagation learning).

In terms of the population size, this has been limited to fifty individuals. The heuristic regarding population size states that it is preferable to limit the population size and increase the number of generations (number of reproduction cycles). Since the genotype is short, limiting the population size will address *non-isomorphism* criteria for efficient genetic coding. Genetic coding fits the criterion of non-isomorphism, if each genotype represents a truly unique phenotype (or solution)

After the population has been generated, each individual's fitness is then evaluated. Here the fitness is evaluated using the mean squared error of the neural network output to that of the desired output on the *validation data set.*

To be complete, the evaluation process for each individual is as follows:

- Using the individual phenotype for the neural network, initialise the weights (see section 4.6.2.3.2) and train the network for 100 epochs using batch supervised learning;

- Repeat the weights initialisation for twenty runs;

- Compute the final average mean squared error of the multiple training runs using the validation data set.

- Sort the individual genotypes in descending order of fitness (the lower the average mean-squared error, the fitter the solution).

The neural training is epoch limited to ensure that the fittest solutions are associated with rapid convergence (or the learning parameters that do not lead to fast convergence are penalised). The population is then reproduced via the most commonly used genetic operators: selection, crossover, and mutation.

4.6.2.2.2 Selection for Reproduction

The selection operator selects genotypes from the population for reproduction. In general, genotypes are randomly selected, with high-fitness genotypes having a higher chance of being selected. For example, the probability of a genotype being selected can be taken proportional to its absolute fitness value (Goldberg, 1989). Another possibility is rank-based selection (Whitley, 1989), in which the selection probability is defined as a linear function of the rank of a genotype in the population according to its fitness. In this research a special case of fitness proportionate selection called roulette wheel selection has been used (Salustowicz, 1995). Individual genotypes are randomly selected for inclusion in the next generation with probabilities proportional to their absolute fitness. However, it is to be noted that the selection method utilised is problem dependent and the probabilistic approach to selection of the fittest genotypes may not show high performance (Tabakov, 2001).

It is common for experimenters to use *elitism*, meaning that the fittest genotype from each generation is carried over into the next generation and replaces the worst genotype of that generation. It has been shown (Matteucci, 2002) that, due to the convergence of the population fitness, *elitism* is not mandatory for the effectiveness of the evolutionary search process. Therefore it has not been applied here.

After selection there exists an intermediary pool of genotypes from which pairs of *parents* can be selected for crossover so as to produce *offspring*.

4.6.2.2.3 Reproduction (Crossover)

Crossover is a genetic operator that combines (mates) two genotypes (parents) to produce a new genotype (offspring). The idea behind crossover is that the new genotype may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution of the intermediary population according to the crossover probability (here set fairly high at 0.9).

Usually two crossover points are randomly chosen at which the two parent genotypes are cut. After parts of both genotypes have been interchanged they are *glued* together to produce two new genotypes, examples of which are Offspring A and Offspring B of Figure 4.6.

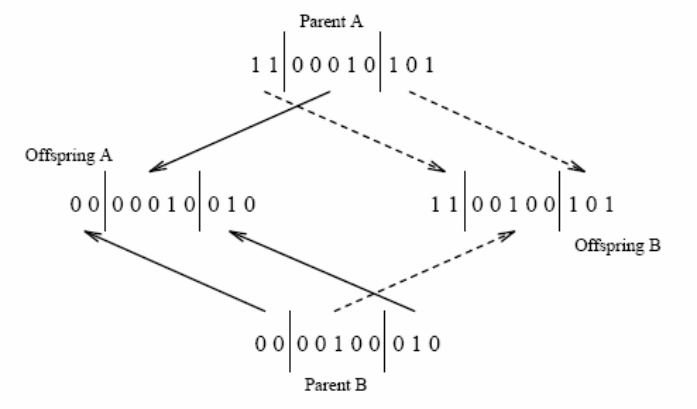


Figure 4.6: Example of Two Point Crossover (Salustowicz, 1995)

A two point crossover operator has been used in this research since multiple cut points do not seem to improve the optimization process (Goldberg, 1989).

It can be shown that this principle of structured information exchange leads to an exponential increase of highly fit pieces of genetic information. Such pieces of short defining length are called *building blocks*. Building blocks can be viewed as coding partial solutions to the problem at hand. These partial solutions are combined by the genetic algorithm to produce good overall solutions. The notion that the combination of fit building blocks leads to better overall solutions is called the *building block hypothesis* (Goldberg, 1989). However, unconstrained crossover may produce non-viable offspring, that is, invalid neural network complexity or learning rate parameters. The prevention approach is employed here, where creation of such offspring is not possible. That is, the ranges of acceptable values are constrained within predefined limits.

Randomness may enter into the genetic algorithm at various points. The initial population is usually randomly generated, selection has a strong random component, crossover is applied with a specific probability, and the cut-points are randomly chosen. A final genetic operator called *mutation*, adds yet another random component to the genetic algorithm.

4.6.2.2.4 Mutation

Mutation is the process of randomly modifying a single part of the offspring genotype string with a certain probability. Mutation may improve the performance of the genetic algorithm by occasionally suggesting a new partial solution not present in the population. This protects the algorithm from accidental, irrecoverable loss of valuable information due, for example, to unfavorable crossovers. The mutation probability of a part of the genotype is usually chosen to be low. This minimizes interference with the main genetic operations (Salustowicz, 1995). With a high mutation rate (>0.5) the genetic algorithm would simply turn into a random search. In this research an accepted and commonly chosen mutation probability of 0.1 has been applied.

The offspring fitness is then evaluated and the offspring is inserted into the population. The process of reproduction (evolution) is continued until the stopping criteria is met.

4.6.2.2.5 Stopping Criteria (Termination)

A specific form of stopping criteria has been used here, namely population convergence. That is, the evolution stops when the population is deemed as converged. The population is deemed as converged when the average fitness across the current population is less than the *threshold* percentage away from the best fitness of the current population. Population convergence can be identified by the fitness starting to repeat and not improve over time. Once the evolution process is terminated, the genotype with the best fitness is taken as the optimal solution for the neural network.

4.6.2.3 Weight Initialisation and Multiple Training Runs

Network weight initialisation has an important influence on whether a network can capture a problem. It is known that with different (random) weight initialisations a network might once find a solution, solve the problem partly (get stuck in a local minimum) or not at all (Salustowicz, 1995; Thimm, 1997). Backpropagation can also fail in very simple cases (Bradley, 1989 and Lawrence, 1997), resulting in a local minimum significantly worse than the global minimum. Further, different learning rate

parameters can produce different training results. Therefore, weight value initialisation and learning rate parameters are two major sources of noise on the fitness evaluation phase of the genetic algorithm.

Such noise can mislead the evolutionary process because, if due to noise, the fitness resulting from genotype G_1 is better than that resulting from genotype G_2, it does not necessarily mean that G_1 will result in an improved model over that resulting for G_2. In order to reduce such noise, it is therefore advisable to do several training runs (n_{train}) with different random weight initialisations. The average results can then be used to estimate the individuals' fitness (Yao, 1997). This advice has been implemented in this research problem. However, it is noted that this dramatically increases the time for fitness evaluation, and can only be accomplished practically using fast processors and efficient code for the neural network development shell.

Flexer (1995) has motivated the necessity for statistical evaluation of neural network models. That study of neural network experiments has shown that the statistical evaluation quality is, in general, rather low. That study has developed the case for the following minimum requirements for proper neural network modelling:

- the use of different training and test sets
- the use of a third independent data set for parameter tuning (that is, the cross-validation set)
- the computation of performance on multiple training runs
- to report the performance mean, variance and confidence intervals over multiple training runs
- to compute a statistical test (t-test) for the comparison of performances

The first three items have already been discussed in detail. Further, Yao (1997) is in agreement with Flexer (1995) in terms of using the mean of the fitness evaluation for multiple training runs. The main issue raised by Flexer (1995) is the need to employ a statistical test (t-test) for comparison of the means of the multiple training runs. Therefore, it is not only whether the one fitness-mean is less than the next fitness-mean, but their comparison also depends on the fitness variance and the number of multiple training runs conducted. That is, one needs to test whether there is a *significant* difference in the means being compared. This requires ANOVA and the Tukey-Kramer procedure for multiple comparison of the means.

Implementation of such a statistical test within the genetic algorithm will place a major computing overhead on the evolution process (which is already taxed with multiple not training runs), and therefore has not been implemented in the genetic algorithm used in this research. However, statistical tests are implemented in order to compare the training performance of the final neural networks trained on multiple runs (non-modular and modular models)

4.6.2.3.1 Number of Multiple Training Runs

Returning to the issue of multiple training runs - how many multiple training runs need to be conducted? It is possible to try to get significant results by conducting more and more training runs, since higher numbers of training runs imply more degrees of freedom and a decrease of the variance. Cohen (1995) points out that this decrease in variance becomes small when more than 20 training runs are being conducted. Therefore, since the architecture and training optimisation of neural networks using genetic algorithms is processor time consuming, the neural networks have been developed with $n_{\text{train}} \leq 20$.

4.6.2.3.2 Weight Value Initialisation

In terms of the multiple training runs, how should the network weights be initialised prior to each training run? Several weight initialisation methods have been suggested by various authors, and a comprehensive review has been done by Thimm (1997). Rumelhart (1986) observed that if the weights are initialised to zero, these have the tendency to assume identical values during training. They therefore propose random weight initialisation to break the symmetry. Further, Lee (1993) showed that the probability of saturated processing elements in multilayer perceptrons increases with the maximum value of the weights. They conclude that a smaller initial weight range increases the learning speed of multilayer perceptrons; however, they do not suggest an optimal initial weight range.

The extensive review by Thimm (1997) concluded that, for higher order perceptrons (of which the modular network is one), there is no standard heuristic for estimating the initial weight value range. However, it is well established that the range should

fall near the dynamical part of the activation function. In this regard, Russo (1991) recommends an empirical formula for the range of the initial weights:

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i} \right) \tag{4.12}$$

where F_i is the fan-in (or total number of inputs) to the processing element, and in this form is used when the activation function is hyperbolic tangent. Since there are six inputs to the first hidden layer, equation (4.12) defines the range to be (-0.4 to +0.4). However, the fan-in on the second hidden layer is not known. Therefore in this research the weight value range has been selected to lie within (-0.5, 0.5) and corresponds to the suggestion of Salustowicz (1995).

Regarding the distribution of the random weight initialisation, Thimm (1997) concluded that the exact distribution has little effect on the optimal learning time of multilayer perceptrons. Russo (1991) suggests a normal distribution. Consequently a normal distribution has been implemented in this research problem for the random weight initialisation with a mean of zero (with the randomisation seeding number being kept constant for all training runs).

Minimum guidelines for proper neural network development (or experimentation) can be divided into how to select and partition the data and how to verify and validate such experiments; the former being a prerequisite for the latter. The data selection and partitioning issues have already been presented (see section 4.2.2). Therefore, the verification and validation methodology is presented in the next chapter, together with the statistical techniques underpinning the results.

4.7 Summary

The following is a summary of the neural network design and development methodology applied in this study:

- NN model development is an iterative process using experimental optimization. This tends to be *ad hoc*. Hence, an evolutionary approach has been proposed as an alternative for the solution space search effectively to address this issue.

- Neural networks are data dependent. The training set size has been defined as needing to be greater than the number of network weight parameters divided by the acceptable error. This heuristic was applied so as to ensure acceptable NN generalisation performance.

- The data has been partitioned into training, cross-validation and test data sets. Where the test data set is truly independent; as per the requirements for executing double-cross-validation of the models. The total data set size is 11,117 exemplars of which 80%, 14% and 6% are respectively the proportion of training, cross-validation and testing data.

- The inputs have been preprocessed so as to be normalized via a zero-mean unit-variance transformation. This was applied to avoid processing element saturation during training, thereby improving training performance.

- Two hidden layers have been selected as a standard, with the first hidden layer performing local problem-feature extraction and the second hidden layer performing global problem-feature extraction.

- Two neural network topologies have been defined, namely, a non-modular feedforward network and a modular feedforward network. A modular network has been proposed because of known advantages such as interference avoidance (spatial and temporal crosstalk) and lower network complexity.

- The effect the learning rate parameter has on backpropagation learning has been described. Correspondingly, the delta-bar-delta learning rule for weight update and learning rate adaptation has been discussed and applied.

- The evolutionary approach to NN development has been defined. The genetic algorithm application strategy has been presented together with a discussion on the genetic operators used. Here, the genetic algorithm has been used to optimize the delta-bar-delta learning rate parameters. It has also been used to optimize the number of weights in each layer. Since the genetic algorithm is better applied for global searches, it has not been used for establishing the optimum weight values (which is a local search problem).

- Weight initialization has a profound impact on whether a NN captures a problem or not. Weight initialization has been defined based on the activation function saturation criteria.

- The networks with optimized layer weight numbers have been trained multiple times (20 times) to ensure that the pathologies associated with NN learning are effectively addressed.

Chapter 5

Model Verification and Validation Methodology

5.1 Introduction

A significant phase of this study involves the verification and validation of the proposed NN models for the diesel engine performance mapping application. Without thorough model verification and validation, there are no grounds on which to place confidence in the study's results. However, model verification and validation is far from straightforward and is often not performed (Robinson, 2004). This is highlighted by Flexer (1995), where a review of many NN modelling and methodology studies showed that proper verification and validation of NN experiments is seldom done.

Model verification and validation can be defined as:

> **Verification** - the process of ensuring that the model construct has been transformed into a computer model with sufficient accuracy (Davis, 1992);
>
> **Validation** - the process of ensuring that the model is sufficiently accurate for its intended application (Carson, 1986).

Here, model verification can be considered a sub-process of model validation.

Experiments in neural network research are required because the methods employed and the data being analysed are generally too complex for formal treatment. For a given problem, there are no formal criteria to decide on which method is the optimal one. There is a vast literature in statistics and computational learning theory that helps with the decision making process. However, the final decision is always made by an empirical check, an experiment; as in any science that needs empirical evaluation of its theories and heuristics (Flexer, 1995).

Standard statistical techniques provide for an accept or reject answer to the question: does significant statistical evidence exist to declare a model invalid, based on the differences between the model prediction and observation? It is important to understand that it is not possible to prove that a model is valid. Validation is more an issue in terms of the confidence that can be placed in the model for its intended application or purpose. That is, the fundamental issue with validation centres on what level of confidence one will have in the model predictions when the model is used on the real world problem for decision making. This is not easily resolved, and Robinson (2004) suggests that the process of validation should be about trying to prove that the model is incorrect, rather than correct. The applied methodology for the model validation is described in this chapter together with the specific performance measures related to neural networks.

5.2 Performance Measures

A network's performance (on training and test data) can have a number of definitions. Two performance measures have been adapted for the verification of the neural networks developed in this study.

5.2.1 Mean Square Error (MSE)

Mean square error (MSE) is a proven measure of quality and performance in neural network research. MSE equals the mean of the sum of the squared deviations from target, that is:

$$\varepsilon_{\mathrm{MS}} = \frac{1}{N}\sum_{i=1}^{N}\left(y_{\mathrm{R}_i} - y_{\mathrm{M}_i}\right)^2 \tag{5.1}$$

where the number of network outputs is equal to 1, N is the number of exemplars in the data set, y_{M_i} the model or network output, and y_{R_i} the desired output of the real world observations. In the models used in this research there is one output, this being the brake specific fuel consumption.

MSE is a valid index for guiding the development of neural network modelling by:

- setting meaningful goals for the modelling process
- monitoring progress towards the goals
- evaluating the differences between models or techniques

However, there are three practical difficulties or disadvantages with the MSE performance measure:

- MSE depends upon unit of measure. There is no point in comparing MSE from different processes that measure in different units. This is not an issue in this research, since the output unit is g/kW.h
- MSE values have no apparent significance except in comparison with previous values from the same process. Some research has stated that the lower the value the better the network. However, low values could be the result of overtraining that result in poor generalisation performance.
- It is not obvious how to set goals for MSE or how to compare the performance of the model with the actual process.

In terms of setting a goal for the MSE of the model, the question that arises is: what is an acceptable value for the MSE? Some experimenters will say the lowest or minimum value or the lowest mean of multiple runs (see section 4.6.2.3). However, how low must low be in order to be acceptable? One needs to apply experiential judgement on the issue in the context of the application requirements for the trained and validated neural network. That is, the MSE goal is application dependent.

These questions and the practical difficulties listed can be addressed by implementing a normalised version of the mean square error.

5.2.2 Normalised Mean Square Error

The normalised mean square error (MSR) is calculated by dividing MSE by the square of the difference between the target and the nearer specification limit (NSL). The nearer specification limit is consistent with the idea that the utility of a model decreases with the square of the deviation from the target value (or real world

observation). That is, it facilitates the setting of an accuracy range (or error range) goal for the performance of the model.

The normalised mean square error using the nearer specification limit has been presented by Battaglia (1996) as:

$$\varepsilon_{\mathrm{MSR}} = \frac{\varepsilon_{\mathrm{MS}}}{\sum_{i=1}^{N} (y_{\mathrm{R}_i} - \mathrm{NSL})^2} \tag{5.2}$$

Now, if the nearer specification limited is defined as:

$$\mathrm{NSL} = (\varepsilon_a - 1)\, y_{\mathrm{R}_i} \tag{5.3}$$

where ε_a is the acceptable error (given as a fraction) on the target value y_{R_i}

then, for a single output, the normalised mean square error is given as:

$$\varepsilon_{\mathrm{MSR}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\left(y_{\mathrm{R}_i} - y_{\mathrm{M}_i} \right)^2}{\left(\varepsilon_a y_{\mathrm{R}_i} \right)^2} \tag{5.4}$$

5.2.2.1 Significance of MSR

The significance of $\varepsilon_{\mathrm{MSR}}$ is that with a neural network (or model) which perfectly simulates the process, both $\varepsilon_{\mathrm{MS}}$ and $\varepsilon_{\mathrm{MSR}}$ will equal zero. If $\varepsilon_{\mathrm{MSR}} = 1$, it is equivalent to a model that produces outputs just at the nearer specification limit. A model that consistently fails to perform within the nearer specification limit will have an $\varepsilon_{\mathrm{MSR}} > 1$.

In the use of MSR as a model performance measure, the primary issue is to establish an acceptable error on the target value.

5.2.2.2 Acceptable Error (ε_a)

In practical applications, the measured specific fuel consumption as calculated from the instruments could have a propagated error in the region of 2.5% of the true value. This value is demonstrated in Appendix A.3.1. using an instrument propagated error calculation and is supported by BS 5514: Part 3 (Reciprocating internal combustion engines: Performance: Specification for test measurements). Since the neural network specific fuel consumption output is to be compared to that of the actual engine specific fuel consumption, it is justifiable to target for an acceptable error on the neural network in the same region as that typically obtainable from the instrumentation. Therefore the acceptable error on the neural network performance has been taken as ±2.5%.

The specification of the acceptable error for the diesel engine performance application allows for the subsequent specification of the mean square error goal (or acceptable error) and specification of the limit on the number of free parameters in the neural network. Now, dealing firstly with the acceptable mean square error.

5.2.2.3 Acceptable Mean Square Error

Although the normalised mean square error (using the nearer specification limit) is more versatile than the mean square error, it is not implemented in most neural network development shells. The two neural network shells used in this study did not implement MSR as defined in equation (5.4). Therefore, the MSR limit using the acceptable error needed to be converted back into a MSE goal as follows:

Applying equation (5.4) to the training data set, and setting $\varepsilon_{\text{MSR}} \leq 1$ and $\varepsilon_a = 0.025$ as acceptable performance criteria, then the limit for the training mean squared error would be $\varepsilon_{\text{MS}} \leq 39.75$ (g/kW.h)^2. Correspondingly, the limit for the mean squared error for the cross-validation and test data sets can be set at $\varepsilon_{\text{MS}} \leq 40.03$ (g/kW.h)^2 and $\varepsilon_{\text{MS}} \leq 35.12$ (g/kW.h)^2 respectively. The use of these limits on the mean squared error performance will be demonstrated during the verification and validation phases of this study.

It is not simply the lowest mean squared error (or best *goodness-of-fit*) that will produce the best performing model on the test data set. The generalization capability of the model on the test data is more important than the best fit to training data. Therefore, Akaike's Information Criteria is used together with the mean squared error to measure the network's training performance.

5.2.3 Akaike's Information Criteria (AIC)

There are no general rules for selecting the number of hidden units in a neural network. Therefore, one requires information criteria to simultaneously account for both the goodness-of-fit of a model and the complexity of the model required to achieve that fit. Model complexity involves both the number of parameters and the interaction (correlation) between the parameters. One such information criteria, namely, Akaike's Information Criteria (AIC) is used for selecting the best fit model among alternative models. AIC is defined by (Akaike, 1973) as

AIC = -2 log (*maximum likelihood*)
+ 2 (*number of independently adjusted parameters*)

When comparing models, the model with the minimum of AIC can be considered as the best approximation of the true model. In the context of neural networks, Akaike's Information Criteria is used to measure the trade-off between training performance and network size. The goal is to minimize AIC to produce a network with the best generalization. For neural network applications, AIC can be approximated by (Webster, 1989):

$$\varepsilon_{\mathrm{AIC}} = N_{\mathrm{train}} \ln\left(\varepsilon_{\mathrm{MS}}\right) + 2S_n \tag{5.5}$$

where S_n is the number of network weights, N_{train} the number of exemplars in the training data set being applied and $\varepsilon_{\mathrm{MS}}$ the mean squared error.

Figure 5.1 plots ε_{AIC} as a function of the NN weights S_n and divides the plot into acceptance and non-acceptance regions. The acceptance region plot is based on the acceptable MSE goal of $\varepsilon_{\text{MS}} \leq 39.75$ and the number of training exemplars of 8855.

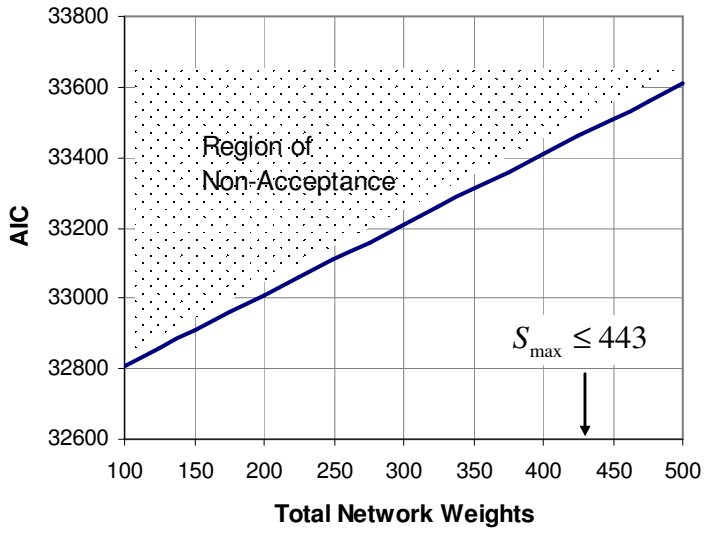


Figure 5.1: AIC Goal as a Function of Network Weights (Training Data Set)

The maximum number of NN weights to meet the error goals is also depicted in Figure 5.1 (see section 5.2.4). Further, note that the plot is specific to the training MSE goal of this study.

Review of Akaike's Information Criteria as defined by equation (5.5) shows that model generalisation capability is a trade-off between training performance and network size. In this study maximum network size needs to be constrained, since network generalisation capability is a primary goal, and a genetic algorithm is being employed for network training optimisation. Therefore returning to the acceptable error goal for the study; the acceptable error must be applied to establish the maximum number of network weights.

5.2.4 Maximum Number of Network Weights Constraint

Limiting the network performance error within the range of the acceptable error ($-2.5\% \leq \varepsilon_a \leq 2.5\%$), allows for the setting of the maximum number of network

weights. The maximum number of network weights ($S_{\max}$) can be calculated using equation (4.1) with the number of training exemplars (N_{train}) and the acceptable error (ε_a) as follows:

$$S_{\max} \leq N_{\text{train}} \cdot 2\varepsilon_a \leq 443 \tag{5.6}$$

where the number of training exemplars is given in Table 4.1. The maximum network weights defined by equation (5.6) sets a good value to which the genetic algorithm can be constrained by using equation (4.2), and thereby limiting the size of the neural network for good generalisation capability.

5.3 Model Verification and White-Box Validation Methodology

Verification and white-box validation is performed on the NN response to desired response when using the training and cross-validation data sets. The process excludes the quadratic model of Golverk (1994a) since it is assumed that this has been performed in that work.

The verification process ensures that during the NN experimental phase there is acceptable performance in terms of error on the training and cross-validation data. This is typically done to keep a check on the development process, for example, the effects of learning rate on solution convergence and overtraining. The verification process also keeps a check on whether there is a network construction coding error that can only be detected during training runs of the NN. This check is typically achieved by placing an error scope on the NN output and tracking the performance during training.

The verification process addresses the following important questions with respect to training performance and NN inputs:

- Is there a significant difference in the mean of the performance measures for the training of the non-modular and modular neural networks? Is there a difference in the multiple training run performance of the networks, namely, the mean squared error and Akaike's Information Criteria using

the training and cross validation data sets? The verification process therefore encompasses a statistical inference procedure using the null and alternative hypothesis statements.

- Is a specific network sensitive to the specified inputs when reviewed using sensitivity analysis? What is the ratio between the variation of the response and that of the specific input? This testing encompasses a white-box validation procedure.

These questions address two aspects of the NN development: training performance points towards the NN having captured the problem and its potential generalisation capability, and since inputs can affect NN generalisation performance, sensitivity analysis provides information to test for the applicability of a NN input.

5.3.1 Verification – Null and Alternative Hypothesis Testing

For the verification of the NN training performance and the comparison of performance between the NN models, the null and alternative hypotheses are stated as follows:

$$H_0: \quad x_A = x_B$$
$$H_a: \quad x_A \neq x_B$$

where x_A and x_B are the parameters of interest from the samples being compared, for example, sample mean.

Where appropriate, the significance of any differences between the models and data sets is determined using a hypothesis test to determine the probability that a given hypothesis is to be rejected or not rejected. The usual process of hypothesis testing consists of four steps (Weisstein, 2004):

1) Formulate the null hypothesis H_0 (commonly, that the observations are the result of pure chance) and the alternative hypothesis H_a (commonly, that the observations show a real effect combined with a component of chance variation);

2) Identify a test statistic (a test to determine the statistical significance of an observation) that can be used to assess the truth of the null hypothesis;

3) Compute the p-value, which is the probability that a test statistic at least as significant as the one observed would be obtained assuming that the null hypothesis was not rejected. The smaller the p-value, the stronger the evidence against the null hypothesis (note that the converse is not true – a higher p-value does not imply that the null hypothesis is true);

4) Compare the p-value to an acceptable significance value α. If $p \leq \alpha$, that the observed effect is statistically significant, the null hypothesis is rejected, and the alternative hypothesis accepted.

For the statistical significance tests conducted in this research the following statistical test parameters have been selected:

5.3.1.1 Confidence Interval and Level of significance (α)

A 95% confidence interval ($\mathrm{CI}_{1-\alpha}$) has been selected for the verification hypothesis testing. Paired sample t-tests were used to test whether the fitness or performance means were different. Therefore, a two tailed level of significance ($\alpha/2$=0.025) has been specified for the tests.

It is recognised that setting the level of significance that partitions the results into significant and non-significant classification is arbitrary and there is considerable debate in the scientific and engineering research community around the issue (see Abelson, 1997). A 95% confidence interval has been selected since it conforms to accepted practice in the applications covered by this study.

5.3.1.2 Model Performance Comparison (Mean and Confidence Interval)

The final training performances of the non-modular and modular network need to be compared based on the training and cross validation-data sets. This is to establish whether there is a difference in training between the non-modular and modular networks. Further, each final neural network and the quadratic model generalisation performance needed to be evaluated against the test data set.

With respect to the neural network training and for each network type, the network with the best evolved fitness (as determined using the genetic algorithm) has its topology fixed (number of weights fixed) and then multiple training runs are conducted and the performance measured. Flexer (1995) suggests that it is not justified to report just the best result of the multiple training runs of a neural network model (with each training run beginning from a different starting point in the weight space). At least the mean of the performance measure over all the runs and the corresponding sample variance (s^2) should be reported to give a better estimate of true performance. It is even better to report the mean over the multiple runs and the corresponding confidence interval that is computed from the sample standard deviation (s).

For neural network result distributions, the distribution of the individual performance results is often of interest rather than the mean performance from a number of trials. Lawrence (1997) has argued that comparisons based on the mean and standard deviation of results can be misleading if the observer assumes the distributions are normal (Gaussian) and in fact are not. Typically, if there are more than thirty multiple training runs and the t-distribution is assumed, then the central limit theorem can be relied on and the distribution of the means can be assumed normal. In this research, due to training optimisation time constraints, the number of multiple training runs conducted was limited. Therefore, a normal distribution cannot be assumed and the recommendations of Lawrence (1997) have been adapted for analysis of the results as follows:

- Use the descriptive statistics - median, interquartile range, minimum and maximum values as well as the mean and standard deviation for interpreting results.

- When plotting results, use box and whisker plots (see Appendix A.3.6 for definitions).

- Test for normality using the Kolmogorov-Smirnov (K-S) test or review the skewness and kurtosis of the distribution.

- Review the confidence interval for the mean.

5.3.1.3 Sample Distribution Measures

Descriptive statistics combined with the box and whisker plots can provide more information about the distribution of the results and, consequently, give insight into the neural network learning process. Further, the use of the parametric tests requires that the normality of the results distributions be reviewed using the measures of skewness and kurtosis, where:

> **Skewness** - a measure of the asymmetry of a distribution. The normal distribution is symmetric, and has a skewness value of zero. As a rough guide, a skewness value more than twice its standard error is taken to indicate a departure from symmetry. That is, the ratio of skewness to its standard error (standard error of skewness) can be used as a test of normality. Therefore, the observer can reject normality if the ratio is less than -2 or greater than +2.
>
> **Kurtosis** - a measure of the extent to which observations cluster around a central point. For a normal distribution, the value of the kurtosis statistic is 0. Positive kurtosis indicates that the observations cluster more and have longer tails than those in the normal distribution and negative kurtosis indicates the observations cluster less and have shorter tails. The ratio of kurtosis to its standard error (named the standard error of kurtosis) can be used as a test of normality. That is, the observer can reject normality if the ratio is less than -2 or greater than +2.

In the statistical tests of this research, a standard error of skewness or kurtosis within the range -1 to +1 will be taken to indicate that the distribution is normal.

When needing to compare the mean performance of two alternative models, what statistical tests need to be employed? There are a whole host of parametric and non-parametric test available based on critical assumptions about the data. However, regarding the performance comparison between two alternative networks and the confidence intervals, the advice by Feelders (1995) and Egmont-Peterson (1994) is to use a t-test, which should be calculated to test the significance of the difference between the performance means.

5.3.1.4 NN Training Performance Comparison t-test

The t-test had been adopted in the analysis of the results to define the confidence interval of the mean as follows:

5.3.1.4.1 Independent Sample t-test:

When comparing the non-modular network performance to that of the modular network, the independent-sample t-test (Appendix A.3.3.1) is used to test the stated hypotheses. This test is done using the training and cross-validation data sets.

5.3.1.4.2 Paired or Related Sample t-test

When comparing the individual neural network performance based on the training data set to that for the cross-validation data set, the paired-sample t-test (Appendix A.3.3.2) is used to test the stated hypotheses.

Hypothesis testing is not performed on the results of the sensitivity analysis, rather the sensitivities are ranked according to their relative magnitude and a decision made from the ranking. Further, a White-Box validation methodology is applied using the sensitivity analysis approach for model validation.

5.3.2 White-box Validation using Sensitivity Analysis

White-box validation is performed by checking that the input-output or input-response mapping behaviour of the NN follows that expected by the domain specific expertise. That is, is the NN response to a specific input true to that of a real or actual diesel engine? Conceptually the process is shown in Figure 5.2.

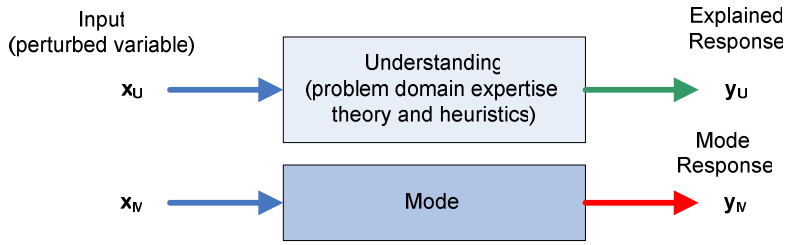


Figure 5.2: White-Box Validation: Comparison with Understanding

Where in Figure 5.2, if y_U is the response that can be explained due to the specific input x_U and y_M is the model response due to the specific perturbed input x_M under the condition that $x_M \approx x_U$, then the white-box validation performed using a variable perturbation methodology addresses the following question:

- Can the NN response to the input variable be understood and explained by the domain expertise, theory and heuristics?

NN models developed that have input-response mapping "not true" in general to actual diesel engine behaviour cannot be stated as valid for the application. This is because there is no basis for interpretation of whether the NN is an accurate representation of the real system; in this case a specified diesel engine. The decision of "not true" or "true" is not statistically tested but rather decided on using individual input-response maps (graphical plots) for each perturbed variable. Note that this specific white-box validation is performed using the training data set.

This white-box validation methodology addresses the typical concerns engineers have as to what the NN is actually doing. Further, this type of validation typically leads to a large improvement in the confidence that can be placed in the model for the specific application.

5.3.2.1 Sensitivity Analysis

As a network is trained, it is advantageous to know the effect that each of the network inputs is having on the network output. Sensitivity analysis methods estimate the rate of change in the output of a model caused by change in the model inputs. It is mainly used to determine which input parameter is more important or sensible to achieve accurate output values, to understand the behaviour of the system being modelled, to verify if the model is behaving as expected and to evaluate the applicability of the model (Saltelli. 2000; Tao, 2000). In the neural network modelling process, it is important to establish whether the network behaviour is somehow related to expected behaviour. That is, can all the observed effects be physically explained? This is the white-box validation phase of the NN development and corresponds to the initial investigation in Chapter 2 as to the understanding of the engine parameters characterising engine performance.

Frey (2001) classify sensitivity analysis methods into three categories: mathematical, statistical and graphical. Mathematical methods assess sensitivity in output values of models to the range of variation of an input. Statistical methods involve running simulations in which inputs are assigned probability distributions and then assessing the effect of variance in inputs on the output distribution. Statistical methods allow one to identify the effect of interactions among multiple inputs – which is statistical experimental design (see Montgomery, 2000). Graphical methods involve information visualization via representation of sensitivity in the form of graphs, charts, or surfaces.

The two methods applied here are the variable perturbation input-response mapping (graphical) and sensitivity analysis.

5.3.2.1.1 Perturbed Variable: Input-Response Mapping

The input-response mapping analysis tests the rate of change in the network output with respect to direct changes in each input. This method adjusts (perturbs) the input values of one variable while keeping all the other variables constant. Plotting the input variable to the network response assists with verifying that the input-response mapping corresponds to the *a priori* information about the relationship. That is, does the input-response mapping correspond to the expected relationship as defined by

known diesel engine operational characteristics and thermodynamic theory? This is a critical phase of the neural network development. Any network not demonstrating the correct or reasonably correct variable input-response mapping must be considered defective (invalid) and therefore rejected for further conceptual design changes, reconstruction and training.

The perturbed variable input-response mapping analysis is conducted using a perturbed data set constructed from the training data set. This is done prior to any network generalisation analysis using the test data set. Should the test data set be used, it would in fact become classified as part of the training and cross-validation data sets and lose its independence from the network training.

The perturbed variable input-response mapping method is defined and expanded in Appendix A.3.2.1 and is executed as follows:

Using the training data set, each input has been varied around its mean by one standard deviation while all other inputs are fixed at their respective means. The network response to each perturbed input variable has been computed for 50 steps above and below its mean. This process has been repeated for each input. The perturbed input-response results are then graphed to investigate the relationship between the specific input and the associated network response.

5.3.2.1.2 Perturbed Variable: Input-Response Sensitivity

The input-response sensitivity analysis estimates the ratio between the variance of the specified input to the variance of the network response to that input. The method is defined and expanded in Appendix A.3.2.2 and is primarily used to determine whether an input should be retained in the neural network model. The analysis also ranks the input sensitivities so that one has a relative view of the input importance on the response. Inputs with low sensitivity or ranking should be excluded or critically reviewed since this would lead to a simpler neural network model without significant penalty in terms of performance.

The previous methodologies dealt with the network performance on the training and cross-validation data sets. The next section defines the black-box validation methodology applied for the test data set.

5.3.3 Black-Box or Solution Validation Methodology

In black-box validation, the overall behaviour or performance of the model is considered. In performing this validation, the methodology is to compare the simulated model response to the *real world* or test data set. This is similar to the verification and white-box validation performed on the NN models during the training phase (since *real world* data was used for training), however, here it is distinct in that no part of the *real world* being compared against has been *seen* previously by the NN model. This is exactly the double-cross-validation method of Mitchie (1994). This methodology could also be termed *solution validation* since it is only performed at the end of the NN experimentation verification and white-box validation phase.

Figure 5.3 shows the black-box validation process. If confidence is to be placed in the model, when it is simulated with the same inputs as the real system, the outputs should be sufficiently similar.

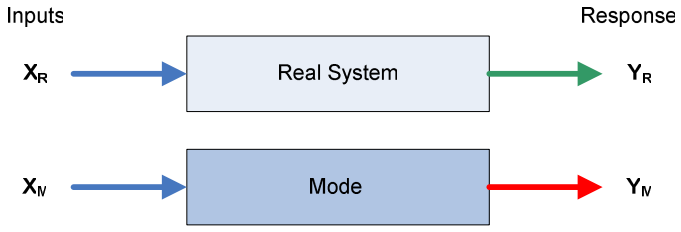


Figure 5.3: Black-Box Validation: Comparison with the Real System (Robinson, 2004)

If $\mathbf{x}_R = \mathbf{x}_M$, with $\mathbf{y}_R$ the respective real system response (or observed response) and $\mathbf{y}_M$ the model response, then the black-box validation null and alternative hypothesis could be stated as

$$\begin{aligned} H_0 &: \quad \mathbf{y}_M = \mathbf{y}_R \\ H_a &: \quad \mathbf{y}_M \approx \mathbf{y}_R \end{aligned} \tag{5.7}$$

Since the model is only an approximation to the actual system, a null hypothesis that the system and model are the same is clearly false and has limited utility in the context of model validation. It is more useful to ask whether or not the differences between the system and the model are significant enough to affect any conclusions derived from the model (Law, 2001; Robinson, 2004; Hills, 1999). That is, one needs to address the alternative hypothesis.

5.3.3.1 Black-box Validation of Model Response Mean

The alternative hypothesis for model mean to test data set mean addresses the following question:

> *Is there a difference in the response means of the non-modular, modular and quadratic method of Golverk when each is compared to the test data (test engine) response mean for the same inputs?*

Two tests are performed for the black-box validation of the mean:

5.3.3.1.1 Test for means within the nearer specification limit

Here the model response mean to the target mean is considered sufficiently similar if the model response normalised mean squared error is less than or equal to 1. Therefore, the alternative hypothesis is accepted as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left(y_{\mathrm{R}_i} - y_{\mathrm{M}_i}\right)^2}{\left(\varepsilon_a y_{\mathrm{R}_i}\right)^2} \le 1 \tag{5.8}$$

where the model response is tested to be within the nearer specification limit defined by the acceptable error goal (or $\varepsilon_{\mathrm{R}} = \varepsilon_a = 0.025$).

5.3.3.1.2 Test for means within the confidence level limits

Here the model response mean to the target mean is considered sufficiently similar if the model response falls within the range defined by:

$$\mathrm{LCL}_{\mathrm{R}} \le (\bar{Y}_{\mathrm{M}} \pm \mathrm{CI}_{\mathrm{M}}) \le \mathrm{UCL}_{\mathrm{R}} \tag{5.9}$$

where the test data set modified lower confidence level is defined as

$$\mathrm{LCL}_{\mathrm{R}} = (1-\varepsilon_{\mathrm{R}})\bar{Y}_{\mathrm{R}} - t_{(n-1),\alpha}\frac{s_{\mathrm{R}}}{\sqrt{n}} \tag{5.10}$$

and the test data set modified upper confidence level is defined as:

$$\mathrm{UCL_R} = (1+\varepsilon_\mathrm{R})\bar{Y}_\mathrm{R} + t_{(n-1),\alpha}\frac{s_\mathrm{R}}{\sqrt{n}} \tag{5.11}$$

given that: $\bar{Y}_\mathrm{M} \pm \mathrm{CI_M}$ is the range of the model response mean (mean and its confidence interval), $\bar{Y}_\mathrm{R}$ the test data set response mean, ε_R the mean estimated propagated error of the test data set responses, $t_{(n-1),\alpha}\frac{s_\mathrm{R}}{\sqrt{n}}$ half the confidence interval of the test data set mean for an α confidence level, response data set size n, and s_R is the sample standard deviation. Also:

$$\mathrm{CI_M} = \varepsilon_\mathrm{M}\,\bar{Y}_\mathrm{M} + t_{(n-1),\alpha}\frac{s_\mathrm{M}}{\sqrt{n}} \tag{5.12}$$

where ε_M is the model mean estimated propagated error (here $\varepsilon_\mathrm{M} = 0.01$).

Therefore, the confidence intervals of the model response mean and the test data response mean are compared when using an acceptable error criterion based on the application (see section 5.2.2.2 and Appendix A.3.1 for the details of what constitutes acceptable error). The neural network is black-box validated by acceptance of the alternative hypothesis as:

$$H_a: \quad \mathbf{y}_\mathrm{M} \approx \mathbf{y}_\mathrm{R} \quad \text{if} \quad \mathrm{LCL_R} \le (\bar{Y}_\mathrm{M} \pm \mathrm{CI_M}) \le \mathrm{UCL_R} \tag{5.13}$$

The validation defined by the statement of (5.14) is severe in that the model mean and its modified confidence interval needs to fall within the upper and lower confidence level of the mean of the test data set.

In summary, the model response mean is validated for the specified application by acceptance of the alternative hypothesis as:

$$
\begin{aligned}
&H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \mathrm{LCL}_{\mathrm{R}} \le (\bar{Y}_{\mathrm{M}} \pm \mathrm{CI}_{\mathrm{M}}) \le \mathrm{UCL}_{\mathrm{R}} \\
&\text{AND} \\
&H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR}} \le 1
\end{aligned}
\tag{5.14}
$$

5.3.3.2 Black-box Validation of Model Response Means at Map Contours

The black-box alternative hypothesis test for the model mean to test data set mean at the performance map contours addresses the following question:

> *Is there a significant difference in the specific fuel consumption contour mean of the response of the non-modular network, modular network and quadratic method of Golverk when each is compared with that of the test data (test engine) response for the same input corresponding to the performance map contours?*

As for the overall mean, two tests are conducted for each contour value:

5.3.3.2.1 Test for contour means within the nearer specification limit

Here the model contour response mean to the target contour value is black-box validated. The model to target response is considered sufficiently similar if the model response normalised mean squared error for the contour is less than or equal to 1. Therefore, the alternative hypothesis is accepted as:

$$
H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR_C}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\left(y_{\mathrm{R}_i} - y_{\mathrm{M}_i} \right)^2}{\left(\varepsilon_a y_{\mathrm{R}_i} \right)^2} \le 1 \tag{5.15}
$$

where the subscript C represents a specified contour value being tested, $Y_{\mathrm{R_C}}$ is the target contour value (single sample – as seen in Figure A.2.4), and ε_{R} the mean estimated propagated error of the test data set responses.

The test, as defined by equation (5.15) is conducted for each contour of the test data set.

5.3.3.2.2 Test for contour means within the confidence level limits

Here the model response means to the test data mean at each contour of test data set are black-box validated. The model response and the target or observed response at each contour are considered sufficiently similar if:

$$\mathrm{LCL}_{\mathrm{R_C}} \le \left(\overline{Y}_{\mathrm{M_C}}\right) \le \mathrm{UCL}_{\mathrm{R_C}} \tag{5.16}$$

where $\overline{Y}_{\mathrm{M_C}}$ is the mean of the model response at the specified target contour value and the lower and upper confidence levels are given by:

$$\mathrm{LCL}_{\mathrm{R_C}} = (1-\varepsilon_{\mathrm{R}})Y_{\mathrm{R_C}} \tag{5.17}$$

$$\mathrm{UCL}_{\mathrm{R_C}} = (1+\varepsilon_{\mathrm{R}})Y_{\mathrm{R_C}} \tag{5.18}$$

Therefore, the model is black-box validated on each target contour value by acceptance of the alternative hypothesis as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \mathrm{LCL}_{\mathrm{R_C}} \le \overline{Y}_{\mathrm{M_C}} \le \mathrm{UCL}_{\mathrm{R_C}} \tag{5.19}$$

The test, as defined by statement (5.19) is conducted for each contour of the test data set.

In summary, the model is black-box validated on each contour of the test data set by acceptance of the alternative hypothesis as:

$$\begin{aligned} &H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \mathrm{LCL}_{\mathrm{R_C}} \le \overline{Y}_{\mathrm{M_C}} \le \mathrm{UCL}_{\mathrm{R_C}} \\ &\text{AND} \\ &H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR_C}} \le 1 \end{aligned} \tag{5.20}$$

The question that arises from this form of validation is: Does the model have to be validated on each contour for the model to be accepted as black-box validated for the application? This question has been addressed as follows:

The model has been validated as a percentage score calculated as the ratio of total validated contours divided by the total possible validations on the contours. The total

possible validations were eight, as determined by the data available for the test engine. The issue is then to specify a limit on the percentage score in order that the model can be validated for the application. Rykiel (1996) suggests that a model is valid should the mean fall within the 95% confidence levels for 75% of the samples being tested. Therefore, the model will be validated by:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{C}} \geq 75\% \tag{5.21}$$

when ε_{C} is given by:

$$\varepsilon_{\mathrm{C}} = 100 \cdot \frac{N_{\mathrm{C}}}{8} \tag{5.22}$$

where N_{C} is the number of times the null hypothesis has been accepted on each contour using (5.20).

The validation defined by the statement of (5.21) is not as severe as that of (5.14) in that only the model means need to fall within the upper and lower confidence level of the mean of the test data set for each contour value.

5.3.4 Response to Target Goodness-of-Fit Analysis

The performance of a trained network can be measured to some extent by the errors on the training, validation and test sets, but it is often useful to investigate the model output or response in more detail. One option is to perform a linear regression analysis and correlation analysis between the neural network model response and the corresponding targets (Hagan, 1996; Reckhow, 1990). This allows for the association or goodness-of-fit to be established between the model response and the required or target response. Note that this analysis is not a new model for the input-response mapping and only uses the network response to the target response for the analysis.

This analysis has been used here as a comparative tool for the two neural network models and the quadratic model of Golverk. The methodology involves the use of the test data set. Therefore it is a subsection of the black-box validation methodology and performed only after the networks have demonstrated acceptable

training performance, and the sensitivity analysis has verified network input-response mapping characteristics.

5.3.4.1 Regression and Goodness-of-Fit Validation

In the modeling results review, not only is the correlation between the model response and target response analysed, also the linear regression line to the target line is graphed with the regression coefficients and the individual confidence interval lines. In contrast to correlation analysis, regression analysis is used primarily for the purpose of prediction. However, in this analysis regression is not used for prediction, but rather used to give further insight (graphical and quantitative) into the neural network response to the target response. The regression analysis methods are defined in Appendix A.3.4, with the application detail as follows:

Typically, the sample regression equation representing the linear regression model would be:

$$\hat{y}_i = b_0 + b_1 x_i \tag{5.23}$$

where $\hat{y}_i$ is the predicted value of y_i for the independent variable x_i and the two parameters b_0 and b_1 correspond to the y-intercept and gradient of the best linear regression relating targets to model outputs. However, in terms of the definitions used in this study, the linear regression line conforming to equation (5.23) would be defined as:

$$y_{\mathrm{M}_i} = b_0 + b_1 y_{\mathrm{R}_i} \tag{5.24}$$

where, as before, y_{M_i} is the model response and y_{R_i} is the target response.

In this study, the coefficient b_1 (gradient) has an interesting physical interpretation. This is since it can indicate the flatness or steepness of the performance map that is generated by the model versus that of the target. That is, if $b_1 < 1$ then the general shape of the model map is flatter than that of the target. Also, if $b_1 \approx 1$ then the general shape is the same as the target, and if $b_1 > 1$ then the general shape is steeper than that of the target.

It should be noted that various authors have several objections to using the regression approach to model validation (Mitchell, 1997; Kleijnen, 1998). The main objection to the regression based validation method is that it violates the statistical assumptions that are the foundation of regression analysis. For example, in standard linear regression, it is assumed that there is no error in the values plotted on the independent variable (in this case the target response). This is certainly not true since the target values have a degree of uncertainty (ε_{R}). Further, the model errors are certainly not independent, and although they are tested for normal distribution, their variances are correlated. However, here the method is used since it provides a simple indication of model response to target response for a three-dimensional surface, which is difficult to interpret without this method. The main issue is to keep in mind the limitations of the method when interpreting the results.

5.3.4.2 Model Least Squares Gradient Validation

Typically in least squares regression, the gradient or beta value can be examined to establish whether it is significantly different from zero. This is accomplished by performing an explicit test of the null hypothesis $H_0: \; b_1 = 0$ by using a form of t-test. However, this form of testing is not helpful in the context of the model validation, since here, there is a specific interpretation of the gradient and the test will typically not be sufficiently severe. With respect to model validation, using the black-box alternative hypothesis, the model is validated as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad b_1 \geq 0.75 \tag{5.25}$$

where b_1 is the least squares regression line gradient.

The coefficient b_0 (y-intercept) has minimal physical interpretation in the analysis other than that it could indicate how well the model can predict the zero specific fuel consumption value. The coefficient is therefore not considered further.

5.3.4.3 Model Goodness-of Fit Validation

Legates (1999) has indicated that as knowledge of physical processes has increased, models have become more complex and often include numerous parameters that are calibrated through optimization procedures, for example, the NN training and GA methodologies used in this study. As a result, statistics other than the standard coefficient of correlation (r) and coefficient of determination (R^2) have been developed to better describe the degree of association between the real world observed data and the model simulated data. The other goodness-of-fit statistics present a viable alternative to the r and R^2 statistics. However, since the biggest problem with all goodness-of-fit measures is interpretation, one must ensure that this is adequately addressed and understood.

Two relative goodness-of-fit statistics (and their modified form) are proposed for use in this study for the black-box validation. These are the Coefficient of Efficiency (Nash, 1970) and Index of Agreement (Willmott, 1981).

5.3.4.3.1 Coefficient of Efficiency:

The Coefficient of Efficiency is defined as:

$$\varepsilon_{\mathrm{E}} = 1 - \frac{\sum_{i=1}^{N} (y_{\mathrm{R}_i} - y_{\mathrm{M}_i})^2}{\sum_{i=1}^{N} (y_{\mathrm{R}_i} - \overline{Y}_{\mathrm{R}})^2} \tag{5.26}$$

where y_{R_i} is the test target response, $\overline{Y}_{\mathrm{R}}$ the mean of the test target response and y_{M_i} the model response to the input of exemplar i.

Physically, ε_{E} is the ratio of the mean squared error to the variance in the observed data, subtracted from unity. The interpretation is thus: if $\varepsilon_{\mathrm{E}} \leq 0$, then the observed mean $\overline{Y}_{\mathrm{R}}$ is a better predictor than the model and, if $\varepsilon_{\mathrm{E}} > 0$, then the model response y_{M_i} is a better predictor than the observed mean. The coefficient of efficiency cannot be interpreted in the same manner as the coefficient of determination (R^2).

Rather it provides a convenient reference point to compare the model predictive capabilities to that for the observed mean. The issue is to establish a level that can validate the model. Coulibaly (2000) indicated that a value less than 0.70 corresponds to a very poor model fit. This leads to a minimum level of $\varepsilon_{\mathrm{E}} \leq 0.85$ being established as a validation limit for this study. Basically, this minimum level is specified at the midpoint between a poor model fit and a perfect model fit.

Therefore using the black-box alternative hypothesis, the alternative hypothesis is accepted and the model is validated as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{E}} \geq 0.85 \tag{5.27}$$

where ε_{E} is the model Coefficient of Efficiency – defined by equation (5.26)

The coefficient of efficiency represents an improvement over the coefficient of determination for evaluation purposes because it is sensitive to differences in the observed and model means and variances. However, due to the squared differences, ε_{E} is more sensitive to outliers. This problem is addressed by the adjusted coefficient of efficiency (Legates, 1999).

5.3.4.3.2 Adjusted Coefficient of Efficiency

The Adjusted Coefficient of Efficiency is defined as:

$$\varepsilon_{\mathrm{E}_1} = 1 - \frac{\sum_{i=1}^{N} \left| y_{\mathrm{R}_i} - y_{\mathrm{M}_i} \right|}{\sum_{i=1}^{N} \left| (y_{\mathrm{R}_i} - \bar{Y}_{\mathrm{R}}) \right|} \tag{5.28}$$

In a similar manner to that of Coulibaly (2000), the acceptance level of the adjusted coefficient of efficiency can be set at $\varepsilon_{\mathrm{E}_1} \geq 1 - \sqrt{(1 - \varepsilon_{\mathrm{E}})}$ for the model validation.

Therefore using the black-box alternative hypothesis, the model is validated by acceptance of the alternative hypothesis as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{E}_1} \geq 0.60 \tag{5.29}$$

where $\varepsilon_{\mathrm{E}_1}$ is the model Modified Coefficient of Efficiency.

Willmott (1981) sought to overcome the insensitivity of correlation-based measures to differences in the observed and model means and variance by proposing the Index of Agreement.

5.3.4.3.3 Index of Agreement

The Index of Agreement $\varepsilon_{\mathrm{IA}}$ is an adaptation of the Nash and Sutcliffe efficiency index. Alteration of the denominator seeks to penalise differences in the mean of the model responses and the mean of the target responses. The Index of Agreement is given by:

$$\varepsilon_{\mathrm{IA}} = 1 - \frac{\sum_{i=1}^{N} (y_{\mathrm{R}_i} - y_{\mathrm{M}_i})^2}{\sum_{i=1}^{N} \left[\left| y_{\mathrm{M}_i} - \overline{Y}_{\mathrm{R}} \right| + \left| y_{\mathrm{R}_i} - \overline{Y}_{\mathrm{R}} \right| \right]^2} \tag{5.30}$$

$\varepsilon_{\mathrm{IA}}$ represents the ratio between the mean squared error and the potential error (PE). Where the potential error is defined as the sum of the squared absolute values of the distances from y_{M_i} to $\overline{Y}_{\mathrm{R}}$, and $\overline{Y}_{\mathrm{R}}$ to y_{R_i}. It represents the largest value attainable for each observation/model pair; or:

$$\varepsilon_{\mathrm{IA}} = 1 - N \frac{\varepsilon_{\mathrm{MS}}}{PE} \tag{5.31}$$

Physically, the interpretation of $\varepsilon_{\mathrm{IA}}$ is similar to that of the normalised mean squared error of Battaglia (1996); where the nearer specification limit is similar to the potential error. Therefore an acceptance level can be set by $\varepsilon_{\mathrm{IA}} \geq 1 - 2\varepsilon_{\mathrm{R}}$

Therefore, using the black-box alternative hypothesis, the model is validated as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{IA}} \geq 0.95 \tag{5.32}$$

where $\varepsilon_{\mathrm{IA}}$ is the model Index of Agreement and $\varepsilon_{\mathrm{R}} = 0.025$.

Like the coefficient of efficiency, the index of agreement is also sensitive to outliers. As a result of this, even poor models can have high values, for example, greater than 0.9 (Legates, 1999). Willmott (1985) therefore proposed a Modified Index of Agreement to address this issue.

5.3.4.3.4 Modified Index of Agreement

The Modified Index of Agreement is defined as:

$$\varepsilon_{\mathrm{IA}_1} = 1 - \frac{\sum_{i=1}^{N} \left| y_{\mathrm{R}_i} - y_{\mathrm{M}_i} \right|}{\sum_{i=1}^{N} \left(\left| y_{\mathrm{M}_i} - \overline{Y}_{\mathrm{R}} \right| + \left| y_{\mathrm{R}_i} - \overline{Y}_{\mathrm{R}} \right| \right)} \tag{5.33}$$

Similar to the Index of Agreement, the validation acceptance limit for the Modified Index of Agreement can be set to $\varepsilon_{\mathrm{IA}_1} \geq 1 - \sqrt{(1 - \varepsilon_{\mathrm{IA}})}$

Using the black-box alternative hypothesis, the model is validated as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{IA}} \geq 0.78 \tag{5.34}$$

where $\varepsilon_{\mathrm{IA}_1}$ is the model Index of Agreement.

5.4 Validation of Performance Maps on Interpolated Data

The black-box validation is extended to validation of the model response via a grid placed over the surface of the test engine performance map. The data points on the grid represent the test engine performance map as horizontal slices (constant brake mean effective pressure) and vertical slices (constant engine speed).

The grid test data are generated from interpolation of the test performance map contour data, and are therefore distinct from the previous test data. Strictly, one cannot use the interpolated test data for validation, since the data itself comes from an interpolation model. However, model validation on this data provides good insight

into the model predictive performance for specific applications. For example, if the engine was used for electrical power generation or certain hydraulic pump/motor applications, the engine would be used at constant speed and variable torque (or variable brake mean effective pressure). In other applications, the engine may be used with constant mean effective pressure (or torque) and variable speed. This review therefore tests for the model validity for these applications.

The interpolated grid test data points were generated using the Delaunay triangulation method of (Barber, 1996) and as implemented by MatLab (The MathWorks, Inc). A total of 1964 interpolated specific fuel consumption data points have been generated from the test data set of 699 points that all lay on the performance map contours. These generated data are then used to produce surface plots.

5.4.1 Validation Criteria: Interpolated Data

Validating the model for on the generated surface, one is concerned about the general shape of the response surface as well as the scale of the error. Therefore, the methodology applied here is: review the shape and error on the performance surface as well as slices through the surface, and review the normalised mean square error on these.

5.4.1.1 Model Validation: Error on Performance Map Surface

A visual inspection is made of the three dimensional engine performance maps generated by each model on the interpolated test data set. These are compared to the target or desired surface in terms of scale and shape. Further, the percentage difference between the model generated surface and the target surface is calculated and plotted on the same coordinates as the test data. The percentage error on the error surface is calculated for each specified fuel consumption data point as follows:

$$\varepsilon_{\mathrm{A}_i} = \frac{\left(y_{\mathrm{RI}_i} - y_{\mathrm{M}_i}\right)}{y_{\mathrm{RI}_i}} \tag{5.35}$$

where $\varepsilon_{\mathrm{A}_i}$ is the absolute percentage error between the target specific fuel consumption y_{RI_i} and the model specific fuel consumption y_{M_i}.

The model is not verified to ε_{A_i}. Rather, as before, each model is black-box validated across the surface of the map by acceptance of the alternative hypothesis as follows:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{RI}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR}} \leq 1 \tag{5.36}$$

Where, $\varepsilon_{\mathrm{MSR}}$ is calculated and based on the interpolated test data set of 1964 exemplars.

5.4.1.2 Model Validation: Horizontal and Vertical Slices Through Map

A visual inspection is made of the model response at specified slices through the engine performance map. These slices are: horizontal slices (constant mean effective pressure) and specified vertical slices (constant engine speed).

Each model is validated across the slices of the map, if the alternative hypothesis is accepted as follows:

Horizontal or constant mean effective pressure slices:

$$H_a: \quad \mathbf{y}_{\mathrm{M_H}} \approx \mathbf{y}_{\mathrm{RI_H}} \quad \text{if} \quad \overline{\varepsilon}_{\mathrm{MSR_H}} \leq 1 \tag{5.37}$$

Vertical or constant engine speed slices:

$$H_a: \quad \mathbf{y}_{\mathrm{M_V}} \approx \mathbf{y}_{\mathrm{RI_V}} \quad \text{if} \quad \overline{\varepsilon}_{\mathrm{MSR_V}} \leq 1 \tag{5.38}$$

where $\overline{\varepsilon}_{\mathrm{MSR_H}}$ and $\overline{\varepsilon}_{\mathrm{MSR_V}}$ are the mean of the normalised mean squared error for the horizontal slices and vertical slices respectively and based on the nearer specification limit, with $\varepsilon_{\mathrm{a}} = 0.025$.

5.5 Summary

This chapter defined the basis of the verification and validation methodology that was applied to the neural network and quadratic models. The main issues centred on the need to verify the neural network model construct using specific performance measures, and the validation of the models for the specific engine mapping application. The following summarizes the pertinent issues dealt with in the chapter:

- Two NN performance measures have been defined, namely, Mean Squared Error (MSE) and Akaike's Information Criteria (AIC). Where, MSE is a valid index for setting meaningful goals for the modelling process as well as monitoring progress towards these goals, and AIC is used to monitor the trade-off between training performance and network complexity.

- Goals for acceptable MSE and generalisation performance error levels have been established using a normalized version of MSE that is based on the Nearer Specification Limit (NSL). The NSL is based on the expected propagated error from the engine and metering instrumentation. From these, an acceptable or sufficient error goal has been set at approximately 2.5%.

- NN sensitivity to each input variable has been tested using raw sensitivity analysis and sensitivity ranking. This is part of the verification phase of the modelling.

- The verification methodology of testing for differences in NN training performances has been defined using independent and related (or paired) sample t-tests.

- A model can never be proved valid; only the confidence in the validity can be established. The verification and validation process centres on attempts to prove the model invalid. Therefore, both white-box and black-box validation processes have been introduced together with twenty three separate tests.

- The white-box validation methodology centres on establishing the input-response mapping of each NN. That is, can the NN model response be interpreted based on domain specific expertise (diesel engine theory and

heuristics)? This validation is based on a perturbed variable data set that has been constructed from the training data set.

- The black-box validation methodology centres on tests to build confidence in the NN model ability to generalize the problem of engine performance mapping. It uses the independent test data set in order to focus on the generalisation capabilities of the models.

- Hypothesis testing with the associated statistical tests have been described. However, in the black-box validation phase, the null hypothesis cannot be defined since it is known *a priori* that the model cannot be the same as the real world engine (a model can only ever be an approximation of the real system). Therefore, an alternative hypothesis has been stated together with conditional tests for acceptance or non-acceptance.

- Conditional tests based on performance error, regression and goodness-of-fit analysis have been presented. These have been presented with specified acceptance limits for the alternative hypothesis.

Chapter 6

Experimental Results

6.1 Introduction

"A mathematical model is neither a hypothesis nor a theory. Unlike scientific hypotheses, a model is not verifiable directly by experiment. For all models are both true and false…. The validation of a model is not that it is "true" but that it generates good testable hypotheses relevant to important problems." (Levens, 1966)

Since it is not possible to prove that a model is valid; the purpose of the validation and verification phase of model development is to attempt to demonstrate that the model is incorrect. Therefore one should rather approach model validation and testing issues in terms of the confidence that can be placed in the model (Robinson, 1999). That is, the more the tests show that the model is not incorrect for the application, the higher the confidence that can be placed in the model for the application. It was with this approach in mind that an analysis of the results of the neural network development methodology as presented Chapter 5 was begun.

In this chapter the training results of the two neural network architectures are verified on the training data set and validated on the test engine data set (engine number 4, Figure A.2.4. in Appendix A.2). Further, the universal performance mapping method of Golverk (quadratic model - see section 2.4.2.2) is validation tested with the test data set. These tests are then used to compared the neural network models to determine whether the neural network methods employed have improved predictive capabilities over the Golverk method.

Analysis of the modelling results substantially relies on the diesel engine theory of Chapter 2, the neural network theory and heuristics of Chapter 3 and 4 and the neural network verification and validation methods of Chapter 5.

The aims of the model performance results analysis are:

1. To verify the construct of the neural network models via training performance measurement;
2. To determine whether there is a significant difference between the training performance of the modular and non-modular neural network when tested on the training and cross-validation data sets;
3. To determine whether the neural network method used for modelling engine performance can be validated for the application. That is, can the model/s developed with the neural network method demonstrate acceptable predictive performance on the test data set;
4. To confirm whether the neural network methods employed have improved predictive performance capability over the quadratic method as proposed by Golverk (1994a);
5. To identify whether further work is needed to produce an acceptable neural network model or select the model that is to be used for actual engine performance prediction.

The analysis of the modelling results has been structured to follow the verification and validation methodology of Chapter 5 as follows.

6.1.1 Model Results Analysis Structure

A) Analysis of the NN Model Training Performance (section 6.2)

The training performance of each neural network was reviewed so as to evaluate whether there was a significant difference between the training performance on the training and cross-validation data sets. This was required in order to verify the neural network models and provide insight into the generalisation capabilities of these models. Further, the training performances of the non-modular and modular networks were compared. This was done to establish whether there were any benefits to employing a modular network approach to the problem (based on the training performance only). This review used the training and cross-validation data sets, and was performed using the methodology as defined in sections 5.2.4 and 5.3.1.

B) Analysis of the NN Input-Response Mapping (section 6.3)

The neural network model responses were investigated using sensitivity analysis. The analysis was part of the verification using white-box validation of the NN models. This checked that the individual input parameter value to the model response had a physical interpretation in terms of the engine performance theory as detailed in Chapter 2. This was required so as to gain confidence in the model's ability to correctly model the response to each input parameter. It also defined and ranked the most important input parameters to the model. This is since, should there be no or limited physical interpretation of the input to response characteristics of the model, the model white-box validity for the application falls away, and hence the support for continuing with the model. This input-response mapping analysis was based on the training data set and used the methodologies defined in section 5.3.2.

C) Analysis of the NN Response Mean to Test Data Mean (section 6.4)

The mean of the responses of each model to the mean of the responses of the test data set were compared. This analysis encompassed both neural networks and the quadratic model. It was the start of the black-box validation phase of the model performance analysis. This analysis presented the first indication of the generalisation capability of each model and employed the methodology as defined in section 5.3.3.1 and section 5.3.3.2.

D) Analysis of the Goodness-of-Fit: Model to Target Response (section 6.5)

The model response to the target response for each model was reviewed. Analysis of specific parameters allowed for evaluation of the goodness-of-fit (or correlation) between the model response and the target response. This analysis used the test or previously unseen data and was part of the black-box validation phase. This analysis was done on all the models and employed the methodology as defined in section 5.3.4.

E) Analysis Using Performance Map (section 6.6)

The previous tests and analyses described above have been based on the model performances when tested at each observed training, cross-validation or test

data. These data are all positioned on the specific fuel consumption contours of the test engine performance map. For model validation, it was necessary to review the model predictive performance in areas of the test engine performance map between the contours. Therefore, the final review was conducted using surface graphs and slices through the surfaces for each model response. The distinct difference between this analysis and the previous analyses was that the test data points were interpolated from the data for each contour. That is, the comparison was between a model response and another model response. This analysis was done on all the models and employs the methodology as defined in section 5.4.

For reference, the summarised results of the verification and validation analyses and tests are shown in Table 6.21.

6.1.2 Model Prefixes and Labels

For ease of presentation and tabulation of the results the models have been given the following prefixes:

Table 6.1: Model Prefixes

Prefix	Description	Reference
M	Modular feed-forward network	Figure 4.3
F	Feed-forward network (non-modular)	Figure 4.2
G	Golverk Universal Performance Mapping Method (quadratic model)	section 2.4.2.1
T	Test engine target response	section 4.2.2 and Figure A.2.4.

The labels on the figures and tables have been defined as listed in Table 6.2.

Table 6.2: Figure Labels and Legends

Label	Description	Reference
model	model association	
response	model response for the test data input	
mse	mean squared error of model output to target output	section 5.2.1
aic	Akaike's Information Criteria	section 5.2.3
cv	Cross-validation response target	section 4.2.2
tr	Training response target	section 4.2.2

6.2 Neural Network Training Verification

Two neural network model architectures have been trained. A modular network (Figure 4.2, or M-model) and a non-modular feed-forward network (Figure 4.3, or F-model). Here, the results of the training are presented in terms of model verification. Further, the question of whether there was a significant difference in training performance between the two neural network models is addressed.

The architecture of the network has been fixed, with the number of weights optimised via an iterative process using genetic algorithms and multiple training cycles. The final network size was then verified based on a fixed number of training exemplars.

6.2.1 Number of Network Weights Verification

As discussed in section 4.2.1.2, network generalisation capability results from a trade-off between training data size, network size and acceptable error. Here, the number of training patterns or exemplars (N_{train}) has been fixed at 8855 (see Table 4.1), and based on the acceptable error, the maximum number of network weights was constrained as $S_{\max} \cong 442$ to satisfy the condition for good generalization performance (see section 5.2.4).

The optimised number of processing elements (not the weight value) in each layer of the non-modular and modular networks is shown in Table 6.3(a).

Table 6.3(a): GA Optimised Number of Weights in Network Layers

Layer	**F-model** (Figure 4.2)	**M-model** (Figure 4.3)
S_1	34	14
S_2	22	6
S_3	1	6
S_4		8
S_5		1

The total network weights have been calculated from the layer weights using equation (4.2). These results are presented in the Table 6.3(b).

Table 6.3(b): Neural Network Model Maximum Weights Verification

Model	**Total Network Weights (S_n)**	**Network Size Criteria Satisfied (5.2.4)**
F-model	449	Yes
M-model	156	Yes

The results of the GA optimization showed that the resulting numbers of network weights have met the criterion for good generalisation performance.

> **Findings**: *Both neural network models have been verified on the maximum number of network weights.*

6.2.2 Training Performance – MSE Verification

A neural network mean squared error performance is influenced by the network weights initialisation values. Therefore, for verification purposes, multiple training cycles were conducted on the final number of weights optimised networks with different (randomly selected) starting conditions (see section 4.6.2.3). The final mean of the error of the multiple training error is then reported with its corresponding confidence interval.

Figure 6.1 gives an overview, using a box-plot, of each neural network's mean square error on the training and cross validation data (see appendix A.3.6 for a description of the box-plot). Table A.4.1 shows the summary statistics for Figure 6.1; for which the mean of ε_{MS} for each model multiple training is summarised in Table 6.4 below.

Table 6.4: Mean of Final Multiple Run Training Mean Squared Error (ε_{MS})

Training Runs N=20	**Training** $(g/kW.h)^2$	**Cross Validation** $(g/kW.h)^2$
F-model	14.18	17.38
M-model	18.03	22.12
Verification Limits	<39.75	<40.03

The verification limit on the mean squared error was set for the training and cross-validation data sets by using the nearer specification limit. The nearer specification limits was based on the propagated instrumentation error as detailed in section 5.2.2.3.

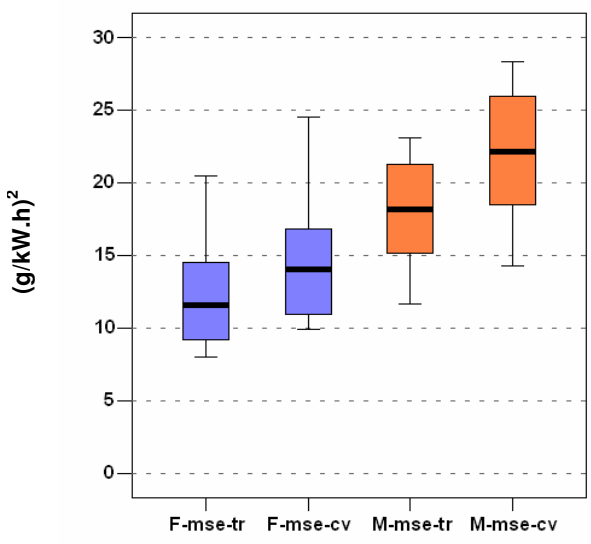


Figure 6.1: Mean of Mean Squared Error on Training and Cross-Validation Data
(Box & Whisker Plot)

From Table 6.4, it is clear that the final mean MSE's on the F-model and M-model were less than the verification limits for both the training and cross-validation data sets.

Findings: *Both trained neural network models are verified based on the mean square error verification limits for both the training and cross-validation data sets.*

Figure 6.1 indicates that the mean of the MSE performance for the F-model was lower than that for the M-model for both the training and cross-validation data sets. It is also apparent that, for both models, the error on the training data was lower than that for cross-validation data. This may indicate that there was a degree of over-training or loss of generalisation capability of both networks.

From these results the questions to be addressed are:

- Was there a significant difference between the training and cross-validation MSE performance for each neural network?

- Was there a significant difference in the means of the MSE performance of the neural network models?

The questions are answered by conducting a t-test to test whether there is a significant difference in the means (see 5.3.1).

6.2.2.1 Individual NN – MSE Performance Comparison

The specific question addressed by this test was: Is there a significant difference between the training and cross-validation MSE performance for each neural network?

The two samples were related since the performance has been measured with the same model, therefore a paired sample *t* test has been applied; the results of which are shown in Table 6.5. The hypothesis testing methodology has been detailed in section 5.3.1 with the null and alternative hypotheses stated as follows:

$$H_0: \quad x_A = x_B$$
$$H_a: \quad x_A \neq x_B$$

where:

Pair 1 - x_A and x_B are the mean of the MSE on the F-model for the training data set and cross validation data respectively, and

Pair 2 - x_A and x_B are the mean of the MSE on the M-model for the training data set and cross validation data respectively.

The two-tailed level of significance was taken as $\alpha/2 = 0.025$.

Table 6.5: Paired Difference t-test – MSE for Training & Cross Validation data on each NN

		Paired Differences							
Pairs		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	p
					Lower	Upper			
Pair 1	F_mse_cv - F_mse_tr	3.20	3.32	.74	1.65	4.75	4.31	19	.00
Pair 2	M_mse_cv - M_mse_tr	4.09	.84	.19	3.69	4.48	21.68	19	.00

Levene's test for equality of variances gave an F value of 1.36 (p=0.25) and 1.44 (p=0.24) for the training and cross validation data respectively, and both the standard error of skewness and kurtosis fall within the range -1 to +1 (see Table A.4.1). Therefore equal variances and a normal distribution can be assumed, thus making the t-test valid in this instance.

If the p value is the probability of obtaining a test statistic equal to or more extreme than the result obtained from the sample data, given that the null hypothesis H_0 is really true, then with $p \le \alpha/2$ the null hypothesis or H_0 is rejected. The results clearly show that, for both pairs, the observed level of significance is less than the level of significance ($p \le \alpha/2$), therefore H_0 is rejected. That is, there is evidence to show that there is a difference in the error performance for the training and cross validation data for each NN model at the $\alpha/2 = 0.025$ level of significance. The confidence interval estimation further shows that the training performance is better

than the cross validation performance (the paired difference confidence interval is a positive range).

> **Findings**: $H_0: \quad x_A = x_B$ *is rejected; for each NN model there is a significant difference between the training and cross validation MSE performance.*

These results therefore support a conclusion that, for both NN models, their generalisation performances may be sub-optimal.

6.2.2.2 Across NN Models – MSE Performance Comparison

The specific question addressed by this test was: Is there a significant difference in the means of the MSE performance of the neural network models?

The two models are not related, therefore an independent *t* test is required; the results of which are shown in Table 6.6. The null and alternative hypothesis are stated as follows:

$$H_0: \quad x_A = x_B$$
$$H_a: \quad x_A \neq x_B$$

where:

Pair 1 - x_A and x_B are the mean of the MSE on the training data set for the F-model and M-model respectively, and

Pair 2 - x_A and x_B are the mean of the MSE on the cross validation data set for the F-model and M-model respectively.

Table 6.6 shows that, for both pairs, the observed level of significance is greater than the level of significance ($p > \alpha/2$), therefore H_0 is not rejected. That is, there is insufficient evidence to show that, when comparing the network training performance, there is a difference in the error performance of the networks at the $\alpha/2 = 0.025$ level of significance (or at the 95% confidence interval of the mean).

Table 6.6: *t* Test for Equality of Means – Across NN models – MSE

		***t* test for Equality of Means**						
Pairs		Mean of Diff	Std. Error of Diff	95% Confidence Interval of the Difference		t	df	p
				Lower	Upper			
Pair 1	F_mse_tr - M_mse_tr	-3.85	2.41	-8.73	1.03	-1.60	38	0.12
Pair 2	F_mse_cv - M_mse_cv	-4.74	3.16	-11.13	1.65	-1.50	38	0.14

The results of Table 6.6 indicate that the different networks tended to train with the same error. Therefore, there seemed to be no apparent benefit in the application of the modular network to the problem. However, this is only based on the training and cross-validation data.

Findings: H_0 : $x_A = x_B$ *is not rejected; between the NN models there was no significant difference on the training and cross validation MSE performance.*

6.2.3 Neural Network Performance – AIC Verification

As for the MSE, verification of the training performance on Akaike's Information Criteria (see 5.2.3) is presented here. Figure 6.2 shows the results of each neural network's mean of Akaike's Information Criteria based on the training and cross validation data. The descriptive statistics for Figure 6.2 are presented in Table A.4.2 with the mean of ε_{AIC} for the multiple training cycles shown in Table 6.7(a).

Table 6.7(a): Mean of Multiple-Run Training Akaike's Information Criteria (AIC)

N=20	**Training (units)**	**Cross Validation (units)**
F-model	23,282	24,989
M-model	25,730	27,542

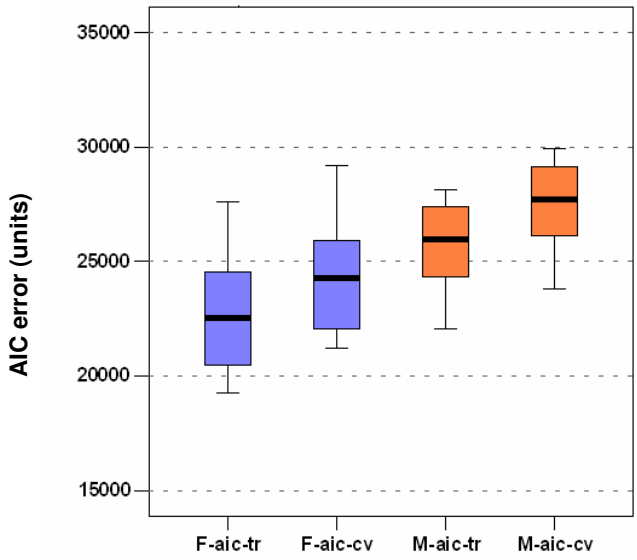


Figure 6.2: Akaike's Information Criteria on Training and Cross-Validation Data
(Box & Whisker Plot)

The verification limits for AIC for each NN model is defined in Table 6.7(b) by using equation (5.5) together with the acceptable mean squared error for each data set, the number of training exemplars of Table 4.1, and the total weights in each network from Table 6.3(b).

Table 6.7(b): Verification Limits - AIC Performance

Data Set	Total Exemplars (N_{tr})	Allowable MSE $(\varepsilon_{\text{MS}})$	**AIC Verification Limits**	
			F-model $(S_n = 449)$	**M-model** $(S_n = 156)$
Training	8 855	40	<33 500	<32 900

Comparison of the results of Table 6.7(a) to the verification limits of Table 6.7(b) indicates that the mean AIC for both trained networks are acceptable. This can also be confirmed using Figure 5.1. That is, the trade-off between the model goodness-of-fit and its size has been verified.

> **Findings**: *The F-model and M-model are both verified on the AIC performance limits.*

Figure 6.2 shows that the mean of the AIC error for the F-model seems lower than that for the M-model for both the training and cross-validation data sets. This tends to imply that the M-model complexity versus the training data set size is worse than that for the F-model. Therefore, if this is true, the F-model should have better

generalisation performance on the test data than the M-model. However, the difference in AIC is not large and the generalization performance still needs to be checked (using the test data set). Further, it seems that for both models, the AIC on the training data is lower than that for cross-validation data.

From these findings, further questions that needed to be addressed were:

- Is there a significant difference between the training and cross-validation AIC performance for each neural network?

- Is there a significant difference in the means of the AIC performance of the neural network models?

As before, the questions were answered by conducting a t-test to test whether there was a significant difference in the AIC performance means.

6.2.3.1 Individual NN – AIC Performance Comparison

The specific question addressed by this test was: Is there a significant difference between the training and cross-validation AIC performance for each neural network?

For each model the means are related. Therefore, a paired sample t test has been performed with the null and alternative hypotheses stated as follows:

$$H_0: \quad x_A = x_B$$
$$H_a: \quad x_A \neq x_B$$

where:

Pair 1 - x_A and x_B are the mean of the AIC on the F-model for the training data set and cross validation data respectively, and

Pair 2 - x_A and x_B are the mean of the AIC on the M-model for the training data set and cross validation data respectively.

Table 6.8 shows the results of the paired sample t-test for each model with $\alpha/2 = 0.025$.

Table 6.8: Paired Difference t-test – AIC for
Training and Cross Validation data for each NN model

		Paired Differences					t	df	p
Pairs		Mean	Std. Dev	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	F_aic_cv - F_aic_tr	1707	290	65	1571	1843	26.29	19	.00
Pair 2	M_aic_cv - M_aic_tr	1811	104	23	1762	1860	77.61	19	.00

Levene's test for equality of variances gave an F-value of 2.76 (p = 0.105) and 2.36 (p = 0.133) for the training and cross validation data respectively. Further, the standard error of skewness and kurtosis fall within the range -1 to +1 (see Table A.4.2). Therefore, equal variances and a normal distribution have been assumed.

The results of table 6.8 show that, for both pairs, the observed level of significance is less than the level of significance ($p < \alpha/2$), therefore H_0 is rejected. That is, there is evidence to show that there is a difference in the training and cross validation AIC performance for each neural network at the 0.025 level of significance; with the AIC performance being found worse on the cross validation data set. The implication is that both models may perform sub-optimally on the test data set.

> **Findings**: H_0 : $x_A = x_B$ *is rejected; for each NN model there is a significant difference on both the training and cross validation AIC performance. The F-model having a lower AIC than the M-model.*

6.2.3.2 Across NN Models – AIC Performance Comparison

The specific question addressed by this test was: Is there a significant difference in the means of the AIC performance of the neural network models?

Here the null and alternative hypotheses are stated as follows:

$$H_0: \quad x_A = x_B$$
$$H_a: \quad x_A \neq x_B$$

where:

Pair 1 - x_A and x_B are the mean of the AIC on the training data set for the F-model and M-model respectively, and

Pair 2 - x_A and x_B are the mean of the AIC on the cross validation data set for the F-model and M-model respectively.

The models are not related, therefore an independent sample t test was performed. Table 6.9 presents the across-model results of the independent sample *t* test for both the training and cross-validation data with $\alpha/2 = 0.025$.

Table 6.9: t Test for Equality of Means – Across NN models – AIC

		t test for Equality of Means						
Pairs		Mean Diff	Std. Error Diff	95% Confidence Interval of the Difference		**t**	**df**	p
				Lower	Upper			
Pair 1	F_aic_tr - M_aic_tr	-2448	970	-4411	-485	-2.52	38	0.016
Pair 2	F_aic_cv - M_aic_cv	-2553	987	-4551	-554	-2.59	38	.014

The results of Table 6.9 show that, for both pairs, the observed level of significance is less than the level of significance ($p < \alpha/2$), therefore H_0 is rejected. That is, there is evidence to show that there is a difference in the AIC error performance across the networks at the 0.025 level of significance and with the F-model having an AIC lower than that for the M-model. The implication is that the F-model would tend to perform better than the M-model on the test data set. However, note that the confidence intervals only marginally do not overlap.

Findings: $H_0: \quad x_A = x_B$ *is rejected; between the NN models there is a significant difference on both the training and cross validation AIC performance.*

6.2.4 Summary of Findings

The following is a summary of the findings of the neural network training performance:

- Both NN models have met the accuracy goals of the study and therefore their training is verified by the tests performed.

- There was no significant difference between the mean square error performance (training and cross-validation data) of the neural networks; that is, both networks tended to learn the training data with a similar scale in error. This is despite the expectation that the modular network should have had a superior ability to capture the problem. A lower mean squared error on the training data was expected for the M-model.

- There was a significant difference between the mean square error of the networks when comparing the training data with the cross-validation data. This indicated that the generalisation performance of both networks may be impaired.

- There was a significant difference in the AIC error performances between the networks, with the F-model having a lower AIC mean than that of the M-model. This indicates that the M-model has a higher complexity (network size) to training set size than that of the F-model. However, since this difference has a confidence interval in the range of 1.4% to 17.6%, the M-model complexity to training set size is only marginally worse than that of the F-model.

- As with the mean square error performance, there was a statistically significant difference between the AIC performances on the training data versus the cross-validation data for each network. The F-model was found to have a lower AIC error than the M-model for both the training and validation data sets.

- Purely on a mean square error and Akaike's Information Criteria performance basis, the F-model demonstrated superior training performance.

- Comparing the NN models' training performances in fact verifies each model and is a partial form of validation. This is since they have closely matching performances on the training and cross-validation data sets. Hence there is a fair degree of confidence in the NN validity prior to entering the validation phase.

Now that the neural network training has been validated, it is critical to test whether the model response is sensitive to the individual input parameters and that the model behaves as expected. The findings of these tests are reviewed and discussed in section 6.3.

6.3 Network Sensitivity Analysis: White Box Validation

In engineering and energy management applications, process models that generate outputs that cannot be interpreted, in general, from the inputs (or fail to behave as expected) should be rejected, modified or further developed and tested. This is fundamentally due to the financial implications of decisions made based on the process model. Poor decisions made on flawed models can result in wasted expenditure. Therefore, a fairly high level of confidence is required in the model prior to its application testing and implementation. Since the domain expert approach has been applied to the development of the NN models of this study, sensitivity analyses as defined in section 5.3.2 have been performed on the neural network models.

This is the NN model verification and white-box validation phase of the model development. Note that sensitivity analysis was not performed on the quadratic G-model; since it was assumed to be validated by that author (see Golverk, 1994a).

Two forms of input sensitivity analysis have been performed:

- The ratio between the specific model input variable variance to its response variance was calculated, with the input perturbed in a specified range with all other inputs held at their mean. The ratios were then ranked. *Therefore, the raw sensitivity is verified*.

- A graphical input-response mapping sensitivity. This was done to establish the relationship between the specific input variable and the model response when the input was perturbed across the perturbation range; with all other variable held at their mean. *The model response is therefore validated*.

The neural network response sensitivities to each of the input parameters as shown in Table 5.10(a) have been tested.

Table 6.10(a): Sensitivity Input Parameter and Perturbation Range

Input Parameter	Description	Symbol	Units	Perturbation Range
x_1	Mean Piston Speed	$\overline{U}_p$	m/s	5.9 – 10.3
x_2	Brake Mean Effective Pressure	P_b	kN/m^2	315 – 692
x_3	Specific Power	$\dot{W}_A$	kW/m^2	540 – 1519
x_4	Piston area to volume displaced	R_{AV}	m^2/m^3	7.1 – 9.9
x_5	Standard Rated Specific Power	$\dot{W}_{A_{\text{rated}}}$	kW/m^2	1719 – 1815
x_6	Standard Rated Specific Fuel Consumption	$\dot{Q}_{c_{\text{std}}}$	g/kW.h	248 - 271

The perturbation range for each input variable has been calculated as per the methodology described in Appendix A.3.2, and is one standard deviation each side of its mean. The statistics for the perturbation range calculation are shown in Table 6.10(b).

Table 6.10(b): Input Variable Statistics and Perturbation Range (Training Data)

Input Variable	x_1	x_2	x_3	x_4	x_5	x_6
Mean	8.1	503.8	1030.0	8.5	1767.5	259.7
Std Deviation	2.2	188.7	489.4	1.4	48.0	11.7
Perturbation:						
Min	5.9	315.1	540.6	7.1	1719.5	247.9
Max	10.3	692.5	1519.4	9.9	1815.5	271.4
Range	4.4	377.4	978.8	2.8	95.9	23.5
Statistics are for sample size of 8855 – training data set						

The questions that the analysis results presented here need to address are:

- Is each neural network model sufficiently sensitive to the selected inputs and does each input warrant inclusion in the model?

- Does each neural network display, in general, the correct input-output mapping for each input variable as expected from the diesel engine theory and known engine behaviour and heuristics?

6.3.1 Model Raw Sensitivity (Verification)

Figure 6.3 shows the response sensitivity to each input for the F-model and M-model respectively. The sensitivity has been calculated as described in Appendix A.3.2.2 and defines the ratio of variance in model response to the variance in the specific input; with all the other inputs held at the mean of their perturbation range. The means and perturbation ranges are shown in Tables 6.10(a) and 6.10(b).

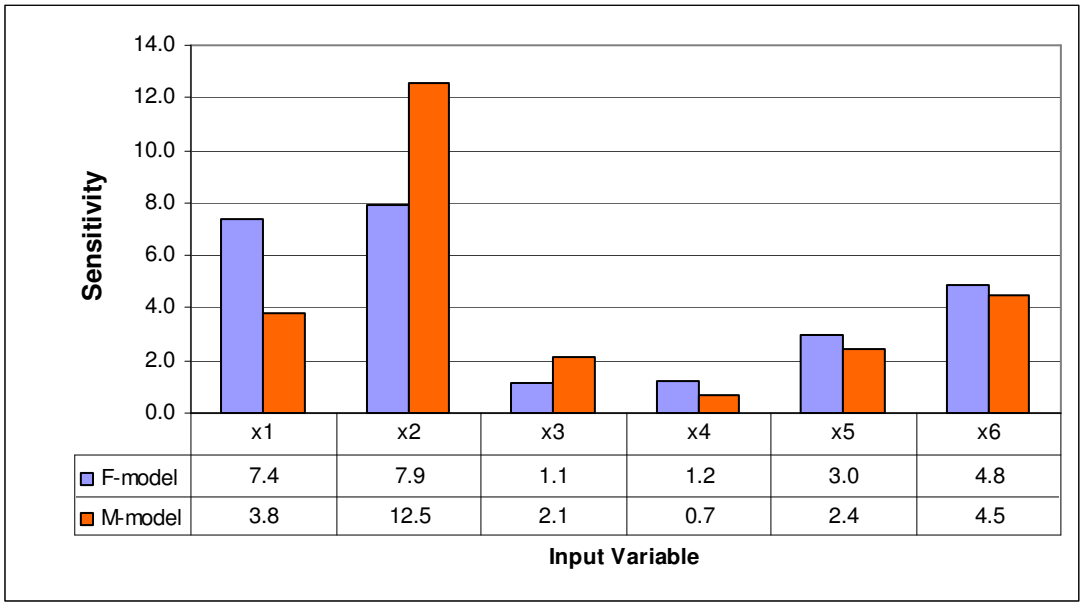


Figure 6.3: NN Model Response Sensitivity to Each Input Parameter

Table 6.11 shows the ranked model sensitivity to the input parameters. For both models the sensitivity to brake mean effective pressure (x_2) is ranked first, and the sensitivity to mean piston speed (x_1) is ranked second.

Table 6.11: Ranked Input Parameter Sensitivity

	x_1	x_2	x_3	x_4	x_5	x_6
F-model	2	1	6	5	4	3
M-model	2	1	5	6	4	3

With reference to Figure 6.3 and Table 6.11, the model sensitivity to each input variable is discussed and verified as follows:

6.3.1.1 Sensitivity to Mean Piston Speed and Mean Effective Pressure

Both NN models are significantly sensitive to changes in mean piston speed ($\bar{U}_p$) and brake mean effective pressure (P_b), indicated by x_1 and x_2 in the Figure 6.3, and have the highest rankings for both models. This sensitivity is expected because the specific fuel consumption of a diesel engine is essentially defined by these two parameters (Haywood, 1988; Harris, 1990; Golverk, 1994a, Celik, 2004). However, where the F-model response sensitivity to $\bar{U}_p$ and P_b are similar (with a difference in sensitivity of 7%), the difference in sensitivity for these parameters of the M-model is substantial (a difference in sensitivity of 229%). The M-model was found to exhibit significantly more sensitivity to changes in P_b than $\bar{U}_p$. That is, the M-model is more sensitive to changes in torque than to changes in engine speed. There is also a substantial difference in the sensitivity to $\bar{U}_p$ and P_b when comparing the models. The M-model response is 58% more sensitive to change in P_b than the F-model response, which in turn is 95% more sensitive to change in $\bar{U}_p$ than the M-model response.

The significance of the described differences is that the gradient of the M-model's specific fuel consumption surface of the performance map will be steeper than the F-model's when moving vertically across the map, and shallower when moving horizontally across the map.

The parameters of the quadratic model of Golverk (1994a/b) indicate that a model should be circa 47% more sensitive to P_b than to $\bar{U}_p$ (see section 2.4.2.2). Comparing the M-model and F-model response to the quadratic model shows that the M-model may be overly sensitive to P_b, but conforms more closely to the findings of Golverk (1994a/b) than the F-model.

Although the neural networks have similar training performance errors, they exhibit distinctly different sensitivities to the mean piston speed and mean effective pressure input variables. That is, they are not approximating the input-output mapping function in the same manner.

6.3.1.2 Sensitivity to Specific Power and Piston Area to Volume Displaced

As seen in Figure 6.3, both models are relatively insensitive to the specific power (x_3) and the piston area to displaced volume (x_4) input parameters; when compared to the mean piston speed and mean effective pressure input parameters. The sensitivity to these input parameters is ranked lowest for both models. This is expected since these are adjusting (or final application calibrating) parameters and not the main drivers of the fuel consumption on an engine. These parameters have been selected to account for subtle characteristic differences between engines.

The analysis established that the M-model is 91% more sensitive to the specific power parameter than the F-model. This is significant in that the M-model seems to have better captured the required relationship between this parameter and the response – see details in section 2.3.1). It was further established that that the F-model response to the piston area to volume displaced parameter is 71% more sensitive than the M-model. These relative differences in response sensitivity between the NN models again imply that the models are approximating the underlying function differently.

It may be argued that, at such low relative model sensitivities to these input parameters, the parameters should be excluded from the neural network input (Sung, 1998). However, since there is typically a relatively small difference in specific fuel consumption between engines of similar ratings, these parameters are retained since they provide a degree of model calibration; which will be beneficial in the practical implementation of the model on an operational engine. However, the cost may be high in terms of the performance reduction on the generalisation test (black-box validation) as implied in the work of Sung (1998).

6.3.1.3 Sensitivity to Standard Rated Specific Power/Fuel Consumption

The sensitivity values and rankings of the standard rated specific power and standard rated specific fuel consumption were as expected. This is since these are major response driving parameters for the NN models. For any engine, these are constants and effectively apply an offset to the NN model response to the P_b and $\bar{U}_p$ parameters.

The F-model was found to be 25% and 7% more sensitive than the M-model for the $\dot{W}_{A_{\text{rated}}}$ and $\dot{Q}_{c_{\text{std}}}$ parameters respectively. This could be due to these parameters compensating for the F-model's lower sensitivity to change in P_b.

> **Findings:** *The raw sensitivity analysis supports the conclusion that the variables warrant inclusion in the NN models. Also, the parameters, although different in sensitivity for each model, were found to be in the anticipated relative proportions to one another. Therefore, both NN models are verified on raw sensitivity.*

Although the model sensitivities to the individual input parameters have been tested, it remains for the actual response to the input parameters to be reviewed. That is, the actual response is reviewed to check physical or expected behaviour of the response (white-box validation). This has been done using the graphical method as categorised by Frey (2001) and is defined by the methodology of 5.3.2.1.1 and Appendix A.3.2.1.

6.3.2 Model Input-Response Mapping (White-box Validation)

The neural network model response is reviewed to determine whether there is some interpretable relationship between the model response to an input parameter and that which is understood from actual diesel engine behaviour in general. This analysis is distinct from pure or raw sensitivity analysis. In this phase of model validation, the physical response is interpreted via an input-response mapping (white-box validation) rather than only the relative magnitude or ranking of the response (verification).

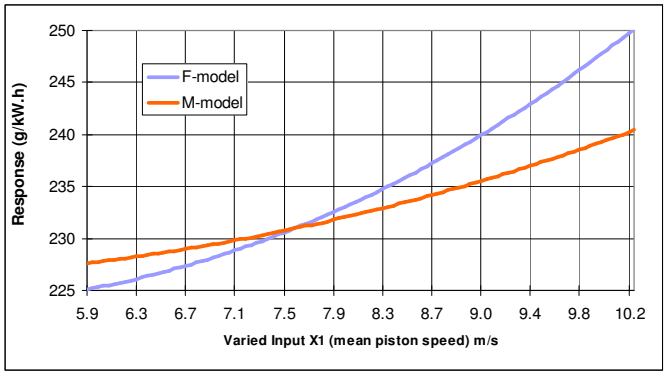
The results of the input-response mapping analysis are shown in Figures 6.4(a) to 6.4(f). These show the F-model and M-model responses to the input parameters that have been varied across their perturbation range. Note that the response axes of the graphs (in units of g/kW.h) do not have the same range and the origin is not at zero.

The findings of the analysis were as follows:

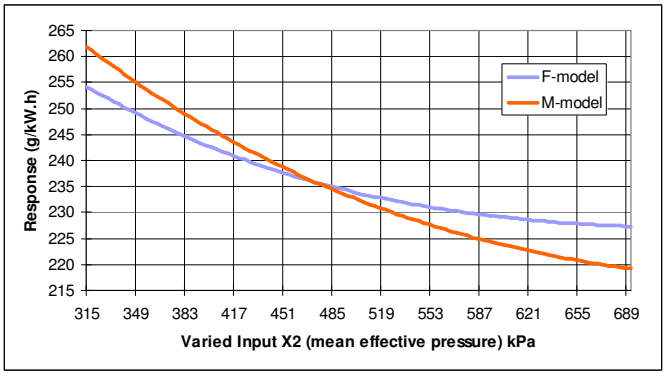
6.3.2.1 Model Response to Input Parameter x_1 - Mean Piston Speed ($\bar{U}_p$)

Figure 6.4(a) shows the F-model and M-model response to a change in mean piston speed ($\bar{U}_p$) in m.s^{-1}. It is clear that, in general for both models, as the mean piston speed increases ($\bar{U}_p \uparrow$) so does the specific fuel consumption ($Q_c \uparrow$). This increase in Q_c with $\bar{U}_p$ is expected behaviour in that, as piston speed increases so does the friction loss or frictional mean effective pressure ($P_f \uparrow$) – resulting in a reduction in mechanical efficiency ($\eta_{\text{mech}} \downarrow$), hence an increase in specific fuel consumption. This effect can be induced from equation (2.13) of section 2.2.3. This effect is identified in operational engines and can be clearly seen in the Figure 6.10(a) which is the performance surface map of the test engine, or Figure 2.6 which is an example of a performance contour map of an engine. In Figure 6.10(a), moving horizontally, at constant brake mean effective pressure ($P_b \rightarrow$), as engine mean piston speed ($\bar{U}_p \uparrow$) increases or rotational speed ($N \uparrow$) increases so does the specific fuel consumption ($Q_c \uparrow$).

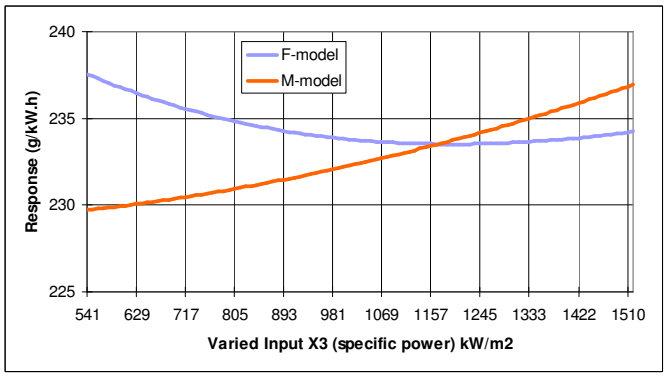
> **Findings:** H_a: $y_{\text{M}} \approx y_{\text{U}}$ *for* $\bar{U}_p$ *is accepted for both NN models. The sensitivity analysis indicates that both model responses exhibit the expected behaviour; with the F-model response being more sensitive than the M-model response with a range of 225 to 250 g/kW.h (11%) for the F-model versus 228 to 240 g/kW.h (5%) for the M-model.*



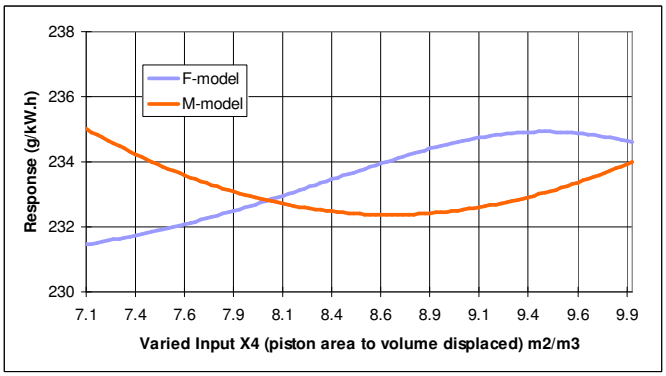
(a) Response to Mean Piston Speed



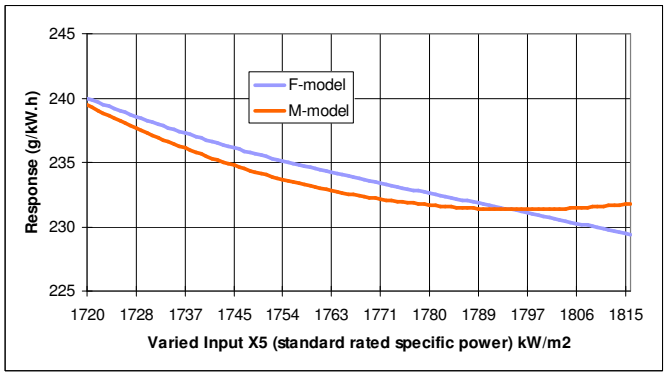
(b) Response to Mean Effective Pressure



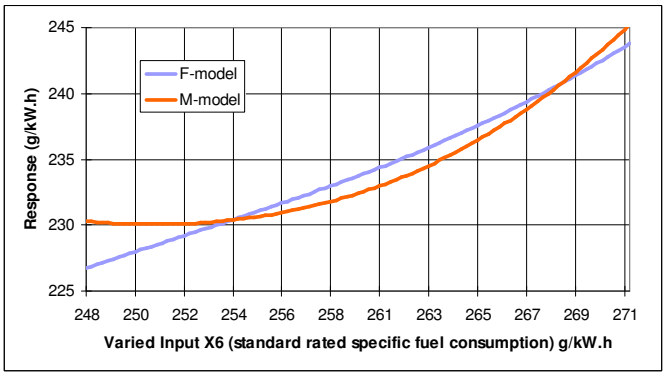
(c) Response to Specific Power



(d) Response to Piston Area to Volume Displaced



(e) Response to Standard Rated Specific Power



(f) Response to Standard Rated SFC

Figure 6.4: Neural Network Response to Varied Input Parameters

6.3.2.2 Model Response to Input Parameter x_2 - Brake Mean Effective Pressure (P_b)

Figure 6.4(b) shows the F-model and M-model response to a change in brake mean effective pressure (P_b) in kPa. It is clear that, in general for both models, as the brake mean effective pressure increases ($P_b \uparrow$) the specific fuel consumption decreases ($Q_c \downarrow$). This decrease in specific fuel consumption for an increase in brake mean effective pressure is expected behaviour, since, as mean effective pressure increases ($P_b \uparrow$) so does the mechanical efficiency ($\eta_{\text{mech}} \uparrow$) – see equation (2.15) of section 2.2.5. This is due to the frictional mean effective pressure remaining nearly constant ($P_f \approx \rightarrow$). The effect of this is an increase in mechanical efficiency ($\eta_{\text{mech}} \uparrow$), and therefore a reduction in the specific fuel consumption ($Q_c \downarrow$). This effect can be induced from equations (2.15), (2.16) and (2.17) of section 2.2.5, and by observing the features of Figure 6.10 (a).

In Figure 6.10(a), moving vertically, that is, at constant mean piston speed ($\overline{U}_p \rightarrow$) or engine speed ($N \rightarrow$), as engine brake mean effective pressure increases ($P_b \uparrow$) the specific fuel consumption decreases ($Q_c \downarrow$). However, this effect is only valid up to a horizontal line passing through the point of minimum specific fuel consumption ($Q_{c_{\min}}$). Increasing P_b from the region of $Q_{c_{\min}}$ along a line of constant engine speed, the fuel-air ratio increases at such a rate that the indicated thermal efficiency decreases ($\eta_{\text{i}} \downarrow\downarrow$) faster than the mechanical efficiency increases ($\eta_{\text{mech}} \uparrow$) - see equations (2.13) and (2.17). This will result in an increase in Q_c ; as can be seen in Figure 6.10 (a).

> **Findings:** H_a : $y_{\text{M}} \approx y_{\text{U}}$ *for* P_b *is accepted for both NN models. The preceding results and analysis indicate that both model responses exhibit the expected behaviour; the F-model response being less sensitive than the M-model response with a range of 255 to 230 g/kW.h (10%) for the F-model versus 262 to 220 g/kW.h (16%) for the M-model*

Before moving onto a review of the responses to the other input parameters, and ignoring their influence for now, it can be seen that combining the model response to $\bar{U}_p$ and P_b will result in a bowl shape. This stretched bowl shape is the distinctive shape of the specific fuel consumption performance map of a normally aspirated diesel engine. The results thus far highlight the importance of these two parameters in the model response. This has been established by these parameters having the highest raw sensitivity ranking as shown in Table 6.11.

6.3.2.3 Model Response to Input Parameter x_3 - Specific Power ($\dot{W}_A$)

Figure 6.4(c) shows the respective F-model and M-model response to a change in specific power ($\dot{W}_S$). From equation (2.20) of section 2.3.1 it has been established that specific power is one quarter of the product of the mean piston speed and the brake mean effective pressure. Correspondingly, the F-model response of Figure 6.4(c) was plotted against one quarter of the product of the mean piston speed and mean effective pressure of Figures 6.4(a) and 6.4(b) respectively.

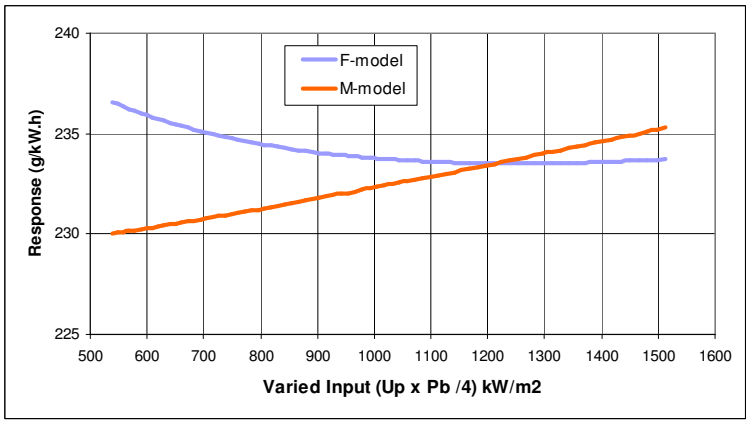


Figure 6.4(g): NN model response to $\left(\bar{U}_p \cdot P_b / 4\right)$

The resulting product, when plotted as shown in Figure 6.4(g), was found to yield a similar shape to that shown in Figure 6.4(c). The same was also found for the M-model.

Typically, as cylinder bore decreases ($b\downarrow$) or the piston area decreases ($A_p\downarrow$) the specific power increases ($\dot{W}_A\uparrow$) – see section 2.3.1. Now, as established by equation (2.24), as the piston area decreases, the cylinder surface area to volume increases and hence the heat lost from the cylinder. The implication is that as specific power increases ($\dot{W}_A\uparrow$), indicated efficiency falls ($\eta_i\downarrow$). Therefore, brake thermal efficiency falls ($\eta_b\downarrow$) and the specific fuel consumption correspondingly increases ($Q_c\uparrow$). Although each model response is confirmed by calculation – as shown in Figure 6.4(g), the F-model response does not follow this expected behaviour at the lower specific power levels (with $Q_c\downarrow$ rather than $Q_c\uparrow$). However, the M-model response does follow this expected behaviour (with $Q_c\uparrow$ as expected) as shown in Figure 6.4(c).

> **Findings:** H_a: $y_{\mathrm{M}} \approx y_{\mathrm{U}}$ *for* $\dot{W}_A$ *is accepted for the M-model and not accepted for the F-model. The range of the response for the F-model was 238 to 233 g/kW.h (2%) and that for the M-model was 230 to 237 g/kW.h (3%).*

The responses of both models are relatively insensitive to the specific power, with rankings of 6 and 5 for the F-model and M-model respectively (see Table 6.11) and therefore have a relatively small influence on the overall output of the model. This input parameter, however, can be deemed to distinctly differentiate the fuel consumption performance of diesel engines.

6.3.2.4 Model Response to Input Parameter x_4 - Piston Area to Volume Displaced (R_{AV})

Figure 6.4(d) shows the respective F-response and M-response to change in the piston area to volume displaced parameter (R_{AV}). From equation (2.24) it is inferred that as piston bore decreases ($b\downarrow$), then piston area to volume displaced increases ($R_{AV}\uparrow$), therefore proportionally increasing the heat lost from the cylinder. The result is that specific fuel consumption increases ($Q_c\uparrow$) with increasing piston area to volume displaced ($R_{AV}\uparrow$). The general relationship between bore size and change in specific fuel consumption is indicated in Figure 2.3. The F-response

displays this expected behaviour with increasing specific fuel consumption for an increasing piston area to volume displaced. The M-response only displays the expected behaviour at the higher R_{AV} values.

> **Findings:** H_a : $y_{\mathrm{M}} \approx y_{\mathrm{U}}$ *for* R_{AV} *is accepted for the F-model and not accepted for the M-model. The F-model response range is small at 232 to 235 g/kW.h (1%) as is that for the M-model at 235 to 232 g/kW.h (1%).*

For both models, greater sensitivity to this input parameter was expected. However the model training data may not have had a sufficiently large selection of engines that would have produced a more diverse range of R_{AV} values for training.

6.3.2.5 Model Response to Input Parameter x_5 - Standard Rated Specific Power ($\dot{W}_{A_{\mathrm{rated}}}$)

Figure 6.4(e) shows the respective F-response and M-response to change in standard rated specific power ($\dot{W}_{A_{\mathrm{rated}}}$). From equation (2.21) it can be inferred that, as the cylinder bore increases ($b \uparrow$), the standard rated specific power decreases ($\dot{W}_{A_{\mathrm{rated}}} \downarrow$). Also, as previously discussed, as the bore increases ($b \uparrow$) the indicated thermal efficiency increases ($\eta_i \uparrow$), correspondingly the brake thermal efficiency increases ($\eta_b \uparrow$). That is, the engine becomes more thermally efficient (this is partly because there is more time for constant volume combustion on larger bore engines – see process b to c in Figure 2.1). The implication is that the specific fuel consumption should increase ($Q_c \uparrow$) with decreasing standard rated specific power ($\dot{W}_{A_{\mathrm{rated}}} \downarrow$). As seen in Figure 6.4(e), both the F-response and M-response exhibit the expected behaviour.

> **Findings:** H_a : $y_{\mathrm{M}} \approx y_{\mathrm{U}}$ *for* $\dot{W}_{A_{\mathrm{rated}}}$ *is accepted for both NN models. The F-response was in the range 240 to 229 g/kW.h (5%) and the M-response was in the range 240 to 231 g/kW.h (4%).*

6.3.2.6 Model Response to Input Parameter x_6 - Standard Rated Specific Fuel Consumption ($\dot{Q}_{c_{\text{std}}}$)

In section 2.3.2 it was shown that the change in nominal rated specific fuel consumption to change in cylinder bore is significant with respect to fuel economy or the specific fuel consumption of the engines. From equations (2.22) and (2.23) it is indicated that the standard rated specific fuel consumption ($Q_{c_{\text{std}}}$) is a function of the cylinder bore size (b), and the difference between the rated specific fuel consumption ($Q_{c_{\text{rated}}}$) and the minimum fuel consumption of the engine ($Q_{c_{\min}}$) – see Figure 2.5. It was also previously shown that the specific fuel consumption increases ($Q_c \uparrow$) with decreasing cylinder bore size ($b \downarrow$) and vice versa. The implication is that, as the standard rated specific fuel consumption increases ($Q_{c_{\text{std}}} \uparrow$), the specific fuel consumption should increase ($Q_c \uparrow$). Figure 6.4(f) shows the F-response and M-response for changes in rated standard specific fuel consumption. Both models exhibit the expected response behaviour. This input parameter is the primary calibrating parameter and is applied in applications for a response offset.

> **Findings:** H_a: $y_{\text{M}} \approx y_{\text{U}}$ *for* $Q_{c_{\text{std}}}$ *is accepted for both NN models. The F-response was in the range 227 to 243 g/kW.h (7%) and the M-response was in the range 230 to 245 g/kW.h (7%).*

6.3.3 Summary of Findings

Analysis of the NN model sensitivity to the six input parameters yielded the following findings:

6.3.3.1 Raw Sensitivity (Model Verification)

- Both the model sensitivities to the input parameters were found to be relatively correct when ranked.

- Both models are most sensitive to the mean effective pressure and mean piston speed parameters. Since these parameters define the general shape of the engine performance map, the models are known to be in general capturing the problem.

- Whereas the M-model was significantly more sensitive to the brake mean effective pressure parameter than the F-model, it was significantly less sensitive to the mean piston speed parameter. The implication is that the general shape of the performance maps for each model would be different.

- The M-model response to mean effective pressure was 229% greater than the response to mean piston speed. This difference in sensitivity for the F-model response was found to be much smaller at 7%. In this respect, the M-model tends to more closely correspond to the findings of Golverk than that for the F-model.

- Both models were relatively insensitive to the specific power and piston area to volume displaced parameters with the lowest ranked sensitivity out of all the parameters. A higher overall sensitivity was expected from these parameters. The M-model was found to be 91% more sensitive to the specific power parameter than the F-model. The F-response was found to be 71% more sensitive to the piston area to volume displaced parameter than the M-model.

- For both NN models the sensitivity to the standard rated specific power and standard rated specific fuel consumption parameters was significant with third and fourth ranking after mean effective pressure and mean piston speed.

- Based on the results of the raw sensitivity, both NN models were considered white-box verified.

6.3.3.2 Input-Response Mapping (White-Box Validation)

- Varying the mean piston speed parameter across its perturbation range yielded the correct response in engine specific fuel consumption - with the specific fuel consumption increasing with increasing mean piston speed. The

NN models therefore behave as expected when based on engine theory related to this input parameter. The models are therefore validated on mean piston speed.

- Varying the mean effective pressure parameter across its perturbation range yielded the correct response in engine specific fuel consumption - with the specific fuel consumption decreasing with increasing mean effective pressure. The models behaved as expected and are therefore validated on mean effective pressure.

- The M-model response to change in specific power was as expected with an increase in specific fuel consumption with an increase in specific power. This was not found with the F-model response. That is, the F-model seemed not to be capturing the problem correctly with respect to the specific power parameter. The sensitivities were in the range 2 % and 3% for the F-model and M-model respectively. The M-model is validated on this input parameter and the F-model is not validated on specific power.

- The model response to the piston area to volume displaced was relatively small at 1% for both models. This could be due to the limited range of the ratio in the training data set. Further, whereas the F-model displayed the anticipated input-response mapping, that for the M-model was only correct at the higher ratios. Therefore the F-model is validated and the M-model is not validated on piston area to volume displaced.

- Although the model sensitivity to the specific power and piston area to volume displaced parameters was found to be relatively small, these parameters have been retained in the NN models so as to be used as model response calibration during actual model application implementation. This is since, when combined with the other inputs, they tend to distinctly characterise an engine's performance.

- The sensitivity response and input-response mapping analyses demonstrated that both neural network models behave, *in general*, as would be expected from the diesel engine and thermodynamic theory underlying engine performance mapping. The results of the input-response lead to both NN models being considered white-box validated.

- The sensitivity analysis showed that the NN models do not approximate the underlying function in the same way. This is despite their relatively close performance error on the training and cross validation data.

- The issues around the internal workings of a neural network being difficult to interpret has been addressed by the sensitivity analysis. This is because the input-response mapping has been defined and interpreted as generally correct. Therefore, there is an acceptable level of confidence in the NN models' validity for the application of this study.

Since the models have thus far been verified and white-box validated, the next stage of the NN model development is to validate whether the neural networks can generalise. That is, demonstrate acceptable prediction of an engine when tested on unseen data – the black-box validation phase.

6.4 Black-Box Validation: Model Response Mean

In this section the means of the model responses are reviewed and compared to the test data response mean for validation purposes. The validation tests performed follow the methodologies described in section 5.3.3.1. Comparison of the model response mean to the target response mean gives a good indication of how well, from a global perspective, the model has managed to predict the required input-response of the test data. Also, in applications where the engine is operated over its entire performance range, this analysis provides an indication of how well the model could predict the engine performance map mean over time.

The analysis here is used to resolve the following question:

> *Is there a difference in the response means of the non-modular, modular and quadratic method of Golverk when each is compared to the test data (test engine) response mean for the same inputs?*

All three models had the same input data, and the mean of the responses to the input data have been calculated for 699 input-response exemplars. A summary of the three model responses and target are shown in the box plots of Figure 6.5 for the test data set. The descriptive statistics for the responses and target are detailed in Table A.4.3 and the summary is shown here in Table 6.12.

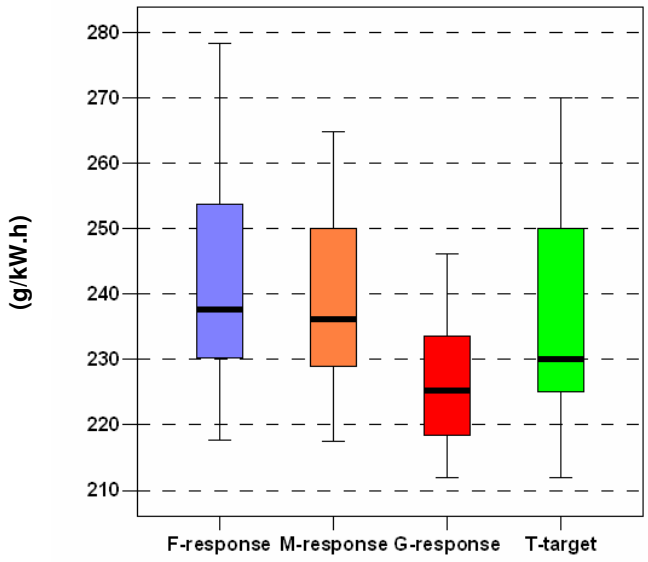


Figure 6.5: Box Plots of the model responses and target for the test data.

From Table 6.12 and Figure 6.5 it is clear that the means and inter-quartile range of the neural network model responses closely resemble the target response. However, the F-response mean is 2.50% higher than the target and that for the M-response 1.09% higher than the target. The G-response mean was found to be 4.23% lower than that of the target.

Table 6.12: Mean & Std Deviation of the Model and Target Responses

	Mean (g/kW.h)	**Std Deviation (g/kW.h)**
F-response	242.34	15.46
M-response	238.83	12.29
G-response	225.89	8.74
T-target	236.51	15.97

6.4.1 Test for Model Means Within the Nearer Specification Limit

The question addressed here was: Does the mean of each model response acceptably predict the mean of the target response on the test data set?

As per the black-box methodology, the null hypothesis is *a priori* rejected and the alternative hypothesis is tested for as follows (see statement (5.14)):

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \mathrm{LCL}_{\mathrm{R}} \le (\overline{Y}_{\mathrm{M}} \pm \mathrm{CI}_{\mathrm{M}}) \le \mathrm{UCL}_{\mathrm{R}}$$

AND

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR}} \le 1$$

The confidence level limits have been tested first and then the normalised mean squared error.

6.4.1.1 Confidence Level Limits Test

Table 6.13 shows the means and adjusted lower and upper confidence levels of the means for each model response. Also shown are the confidence level limits for

model validation. The confidence levels shown in Table 6.13 are based on ε_R set at 0.025, ε_M set at 0.01 as per Appendix A.3.1, and applying equations (5.9) to (5.12).

Table 6.13: Black-Box Test:: Means and Confidence Levels

Sample Size = 699	**Sample Mean (g/kW.h)**	**Confidence Level of Mean (g/kW.h)**	
		Lower Level	**Upper Level**
T-target	236.51	229.41*	243.60*
F-response	242.34	238.77+	245.91+
M-response	238.83	235.52+	242.13+
G-response	225.89	222.98+	228.8+

* confidence level limits of validation acceptance - LCL_R and UCL_R

+ Adjusted confidence level $\overline{Y}_M - \mathrm{CI}_M$ and $\overline{Y}_M + \mathrm{CI}_M$

As seen from Table 6.13, it was found that the F-response upper confidence level falls outside the confidence level limits of the target mean. That for the M-response is within the limits. The G-response lower and upper confidence levels fall outside the confidence level limits.

So as to further clarify the findings, Table 6.14 shows the results of Table 6.13 in terms of the percentage error between the model response mean and the target response mean. Further the acceptance/non-acceptance of the alternative hypothesis is clarified in Figures 6.6 (a) to 6.6(c) for each model (the actual response to target error is also shown for each data point of the test data set).

Table 6.14: Model Response Mean to Target Mean: Percentage Difference

	% Difference Mean	**% Difference in Confidence Level**	
		Lower	Upper
T-target	0	-3.00	3.00
F-response to T-target	2.47	0.96	3.98
M-response to T-target	0.98	-0.41	2.38
G-response to T-target	-4.49	-5.72	-3.26

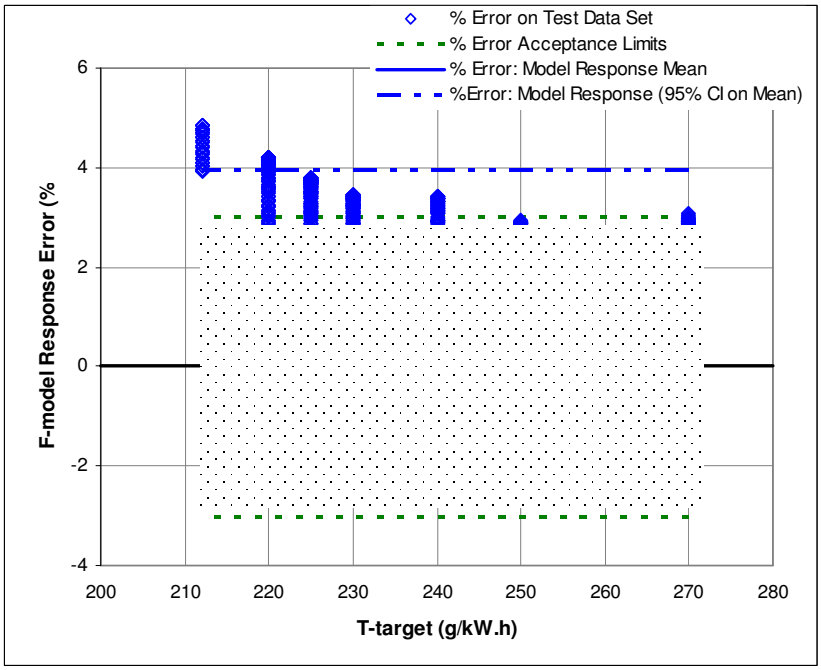


Figure 6.6(a): F-Model Response to Target Response Percentage Error

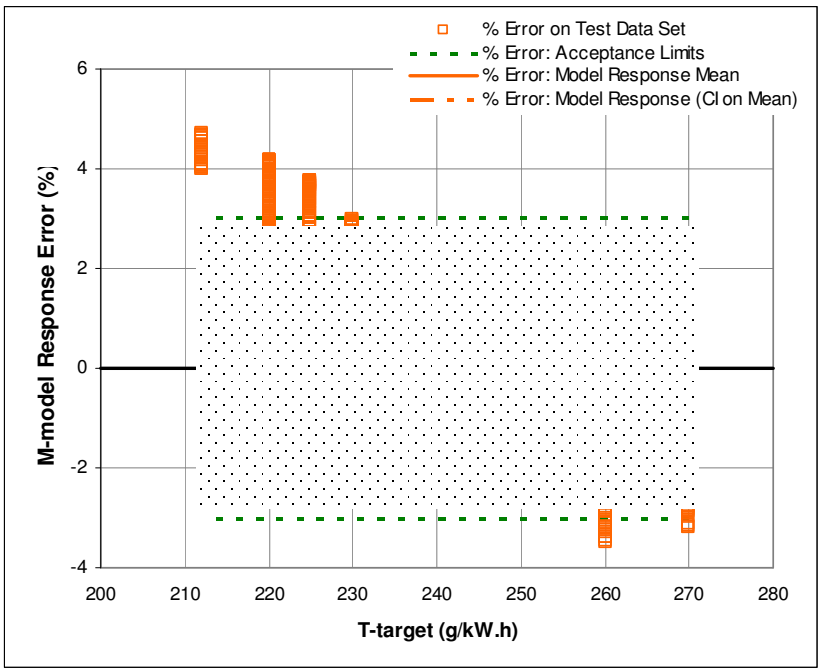


Figure 6.6(b): M-Model Response to Target Response Percentage Error

Review of the confidence levels of Tables 6.13 and 6.14 as well as Figure 6.6(b) indicate that the M-model is validated using the alternative hypothesis acceptance criteria. Further, the M-model response shows an exceptionally small error (and confidence interval) when considering that it is predicting the mean of the test data set (an unseen engine during training).

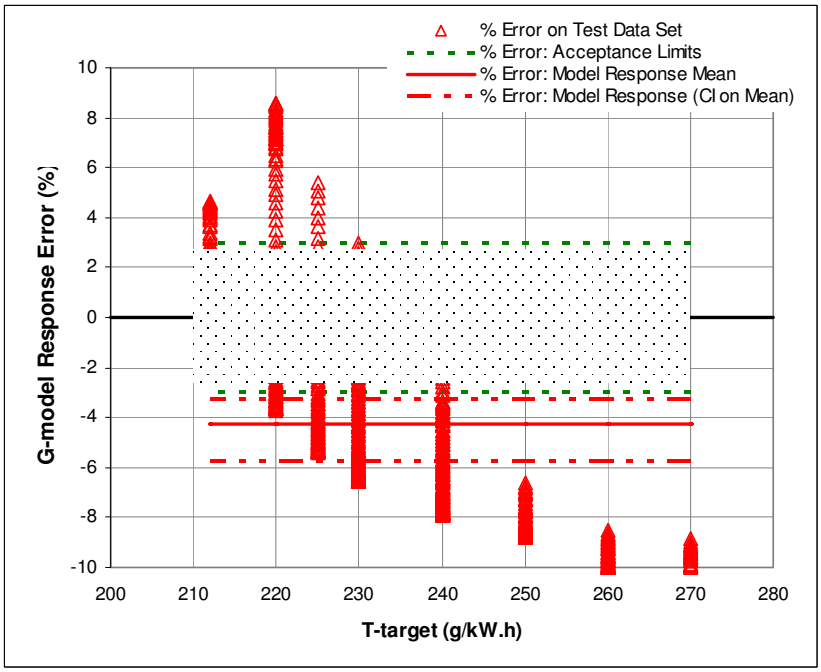


Figure 6.6(c): G-Model Response to Target Response Percentage Error

Regarding the scale of the error for the G-response to target and the confidence interval, one may deem the model unacceptable for application; as is seen by the test results of Table 6.13 and 6.14 and Figure 6.6(c). However, one needs to bear in mind that this can be considered fair model predictive performance considering that the model input parameters are only the normalised mean piston speed, mean effective pressure and minimum specific fuel consumption.

Findings: $H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \mathrm{LCL}_{\mathrm{R}} \le (\bar{Y}_{\mathrm{M}} \pm \mathrm{CI}_{\mathrm{M}}) \le \mathrm{UCL}_{\mathrm{R}}$ *is not accepted for the F-model, is accepted for the M-model, and is not accepted for the G-model. Therefore, only the M-model response mean is black-box validated. However, the F-model mean error was within the acceptable error for the application and only its upper confidence level of the mean was marginally outside that of the limits.*

6.4.1.2 Normalised Mean Square Error Limits Test

Table 6.15 shows the normalised mean squared error on the model response to target response for the test data set. The values are estimated using equation (5.4).

Table 6.15: Normalised Mean Squared Error: Model to Target Response

$\varepsilon_{\mathrm{MSR}}$		
F-response	**M-response**	**G-response**
1.189	0.869	6.652

The results show that the alternative hypothesis for the $\varepsilon_{\mathrm{MSR}}$ criterion is only acceptable for the M-response.

Findings: $H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR}} \leq 1$ *is accepted for the M-model only.*

The M-model was the only model validated on the black-box validation criterion for acceptance of the mean. Also, although the F-model only marginally failed acceptance on the confidence limits, it did not perform well on the $\varepsilon_{\mathrm{MSR}}$ criterion.

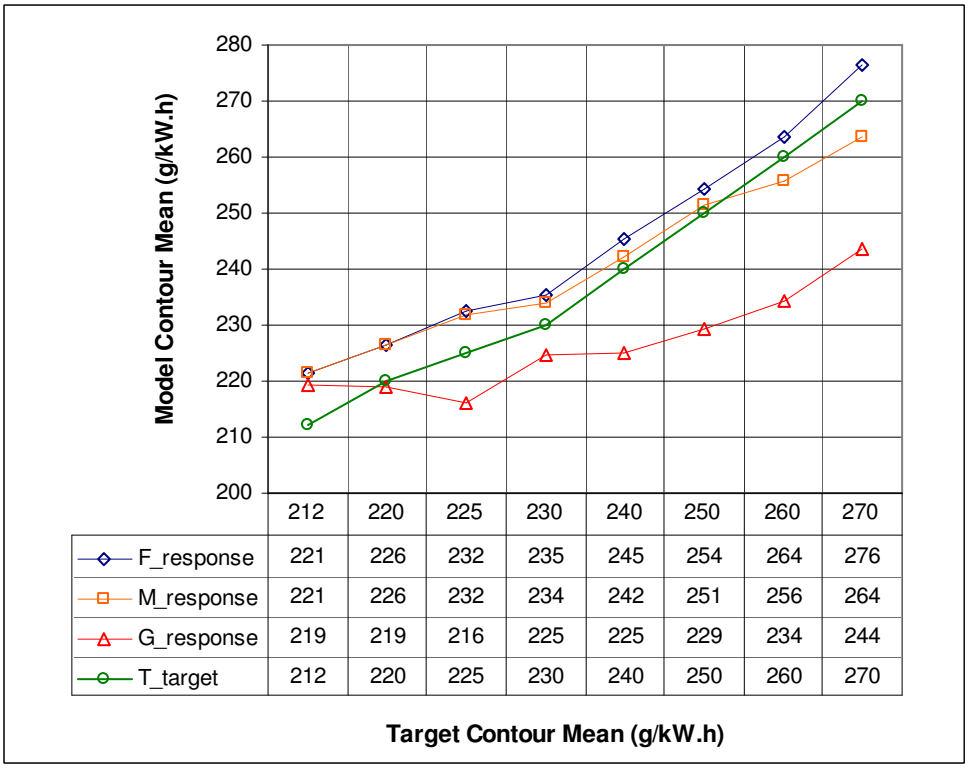
An independent sample t-test found that the F-model had a significantly better AIC performance on the training and cross-validation data (see 6.2.3.2). That result suggested that the F-model (non-modular network) should have had a better generalisation performance than the M-model (modular network). However, the findings of the response mean black-box validation showed the M-model response outperforming the F-model response on the test data set. This finding hints towards the anticipated superior generalisation ability of a modular neural network over a non-modular neural network; even when the training of the non-modular network seemed superior. This result further demonstrates the power and necessity for having a truly independent test data set as per the double cross validation method of Mitchie (1994).

6.4.2 Test for Model Contour Means Within NSL

Further to investigating the overall mean of the model, one is interested in determining how well the models can predict the target response in different areas of the engine performance map (or at different engine operating points). Since the only test data available all lay on constant brake specific fuel consumption lines (contours), the following analysis focuses on comparing the model response mean to

the target response for selected contours on the test data or test engine performance map.

Each model response mean for each contour of the test data set were calculated. There were eight contour means tested and the model response means for these are shown in Figure 6.7; the descriptive statistics for which are presented in Table A.4.4.



	212	220	225	230	240	250	260	270
F_response	221	226	232	235	245	254	264	276
M_response	221	226	232	234	242	251	256	264
G_response	219	219	216	225	225	229	234	244
T_target	212	220	225	230	240	250	260	270

Figure 6.7: Model Mean Response at Target Contours

Validation takes on a similar form to the black-box validation of the overall mean with the null hypothesis being *a priori* rejected, and the alternative hypothesis tested for as follows- see section 5.3.3.2 and equations (5.15) to (5.22):

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \mathrm{LCL}_{\mathrm{R_C}} \le \left(\bar{Y}_{\mathrm{M_C}}\right) \le \mathrm{UCL}_{\mathrm{R_C}}$$

AND

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR_C}} \le 1$$

FOR

$$\varepsilon_{\mathrm{C}} = 100 \cdot \frac{N_{\mathrm{C}}}{8} \ge 75\%$$

Table 6.16 shows the target contour mean, lower confidence level and upper confidence level together with the model response mean at each contour, the validation count N_C and percentage score ε_C. The table is based on $\varepsilon_R = 0.025$.

Table 6.16: Contour Acceptance Levels and Model Means with Validation Score

T-target			**F-response**		**M-response**		**G-response**	
Mean Y_{R_C}	Lower Conf. Level LCL_{R_C}	Upper Conf. Level UCL_{R_C}	Mean $\bar{Y}_{M_C}$	MSER ε_{MSR_C}	Mean $\bar{Y}_{M_C}$	MSER ε_{MSR_C}	Mean $\bar{Y}_{M_C}$	MSER ε_{MSR_C}
212	206.7	217.3	221.3	3.10	221.2	3.03	219.7	2.19
220	214.5	225.5	225.7	1.48	225.2	1.29	223.3	3.60
225	219.4	230.6	232.1	1.64	231.2	1.333	219.3	2.32
230	224.3	235.8	236.0	1.15	234.1	0.67	221.5	3.36
240	234.0	246.0	245.8	1.03	241.7	0.36	225.2	6.46
250	243.8	256.3	255.0	0.74	250.6	0.06	229.4	10.96
260	253.5	266.5	263.4	0.313	255.2	0.67	234.8	15.03
270	263.3	276.8	276.6	1.02	263.4	0.97	243.7	15.22
N_C			4	2	7	5	1	0
ε_C			50%	25%	87.5%	62.5%	12.5%	0%

The alternative hypothesis cannot be accepted for the models, since Table 6.16 shows that all the models fail the validation criteria. However, the M-model has a response mean within the required contour confidence levels with a score greater than 75%. Therefore, the M-model is considered partially validated on the acceptance criteria.

The results of Table 6.16 are shown in Figures 6.8(a), 6.8(b) and 6.8(c). These plot the contour mean of each model response error percentage against the target contour mean. The plots show the model response modified 95% confidence interval of the mean as well as the acceptable lower and upper confidence level acceptance limits on the error.

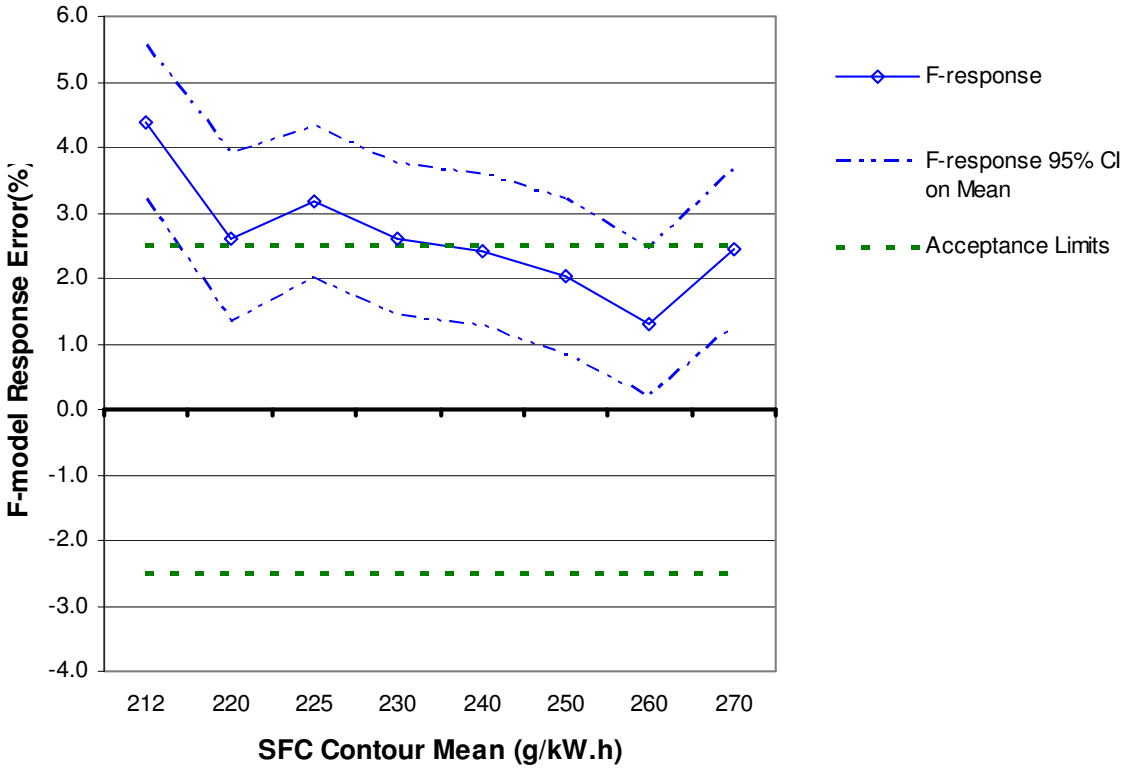


Figure 6.8(a): F-model Response Error on Target Contour Mean

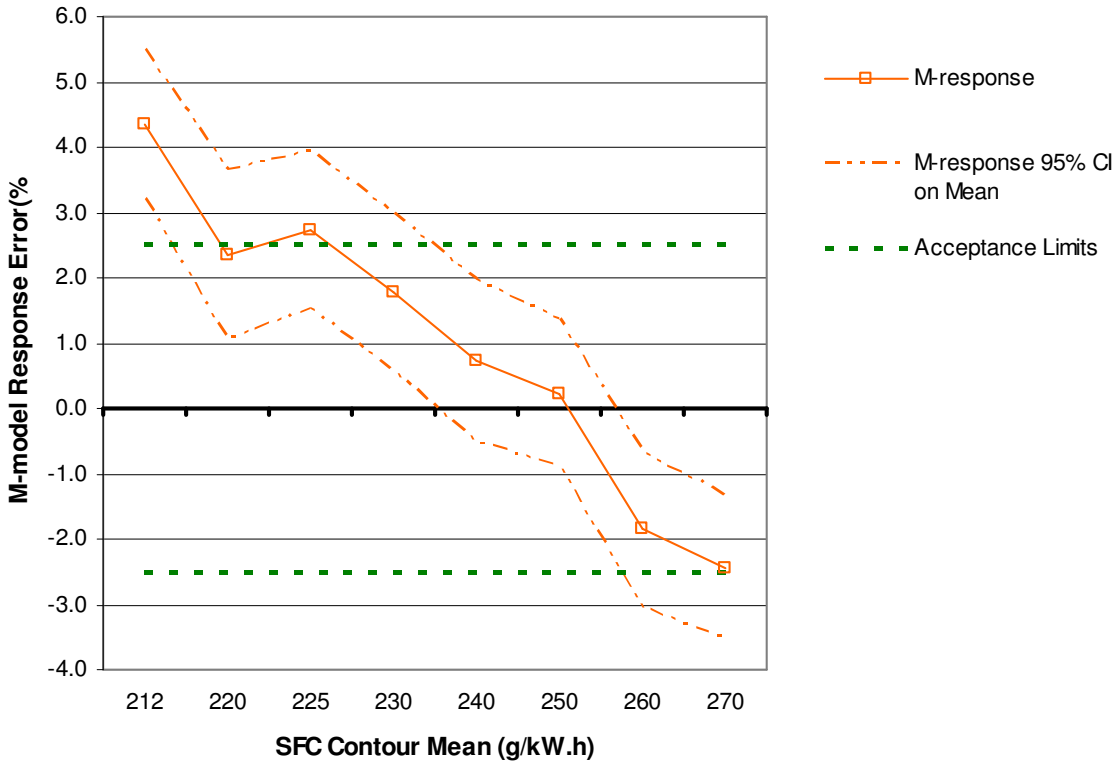


Figure 6.8(b): M-model Response Error on Target Contour Mean

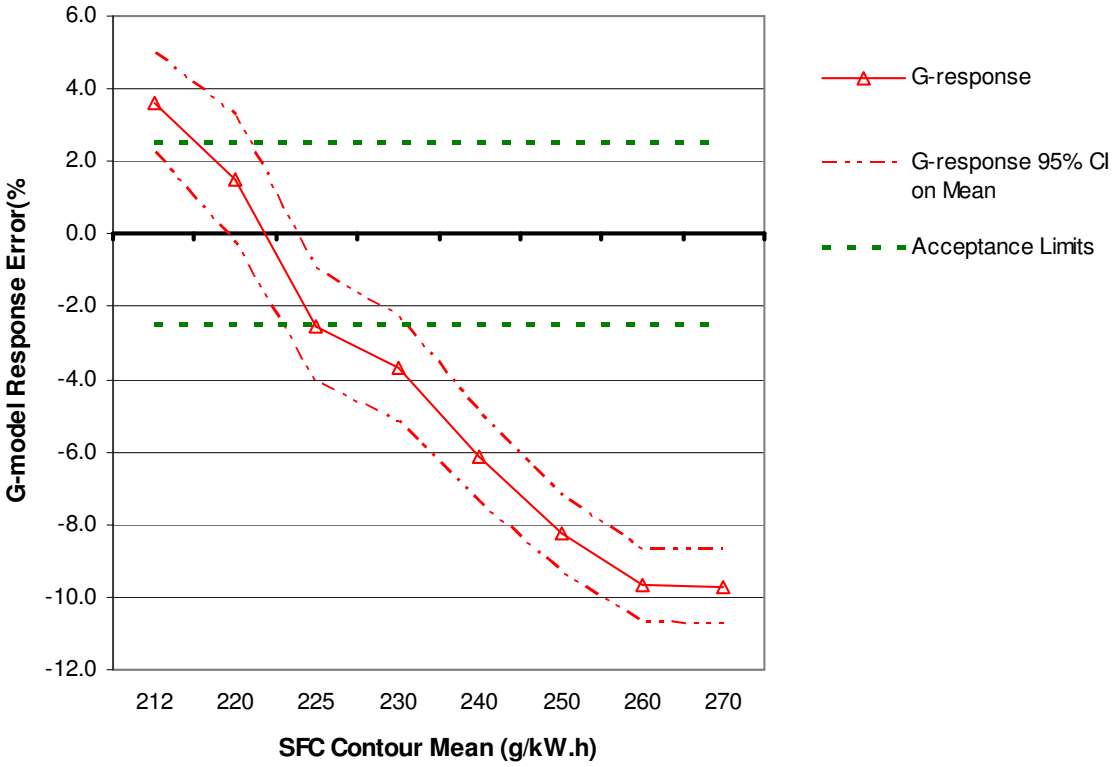


Figure 6.8(c): G-model Response Error on Target Contour Mean

Figures 6.8(a) to (c) show that all three models are relatively poor predictors of the lowest (or minimum) specific fuel consumption of the test engine performance map. Both the F-model and M-model demonstrate fairly good contour mean prediction across the upper lower to high range of the test engine performance map. The confidence intervals on the M-model means overlap those for the test contour means across a significant portion of the performance map. Regarding the G-model, the contour value predictive capability is overall poor. Therefore, it has a low confidence attached to its predictive capabilities when considering the contour response means.

Findings:

Review of Table 6.16 and Figures 6.8 (a), 6.8(b) and 6.8(c) indicate the following:

F-model response: *the F-model is not validated since the validation percentage score of 50% is less than 75%. That is, only four of the eight model response values fell within the confidence levels of the contour target mean. Further, the normalised mean squared error is only within the nearer specification limit for two contour target values (250 and 260 g/kW.h).*

M-model response: *the M-model is validated on the contour mean; with a validation percentage score of 87.5% which is greater than 75%. That is, seven of the eight model response means were within the confidence levels of the contour target values. However, the model is only validated on the 230, 240, 250, 260 and 270 g/kW.h contours where the normalised mean squared error is less than or equal to one. Therefore, the M-model is only partially validated on the contour response means.*

G-model response: *the G-model is not validated since the validation percentage score was 12.5%. Further, the normalised mean squared error was outside the nearer specification limit for all the contour values.*

The validation was fairly stringent in that, the model response mean must fall within the confidence interval of the contour target mean and the normalised mean squared error must be less than the nearer specification limit. Thus, due to its validation score, the M-model is the only model validated on the contour means for the application.

Now that the model response means analysis has been reviewed, in the section that follows, the model response to target response goodness-of fit is reviewed using a number of tests for the alternative hypothesis.

6.5 Goodness-of-Fit Analysis: Model to Target Response

In this section the focus of the analysis moves from the overall mean onto analysis of the model response goodness-of-fit to the target response. The question that needs to be addressed here is: How well does the model response correlate to the target response on the test data set? This question is addressed by employing the validation methodology as defined in section 5.3.4 using the following black-box validation criteria:

6.5.1 Model Validation Criteria

The models are to be validated on acceptance of the alternative hypotheses as:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad b_1 \geq 0.75$$

AND

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{E}} \geq 0.85$$

AND

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{E}_1} \geq 0.60$$

AND

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{IA}} \geq 0.95$$

AND

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{R}} \quad \text{if} \quad \varepsilon_{\mathrm{IA}_1} \geq 0.78$$

Figures 6.9(a), 6.9(b) and 6.9(c) show the regression plot for the model response (dependent variable) versus the target response (independent variable) for the F-model, M-model and G-model respectively. There were 699 data points in the test data set. The least-squares linear regression line is shown and this is banded by the 95% individual prediction interval lines. The condition where the model response is equal to the target response is shown for reference.

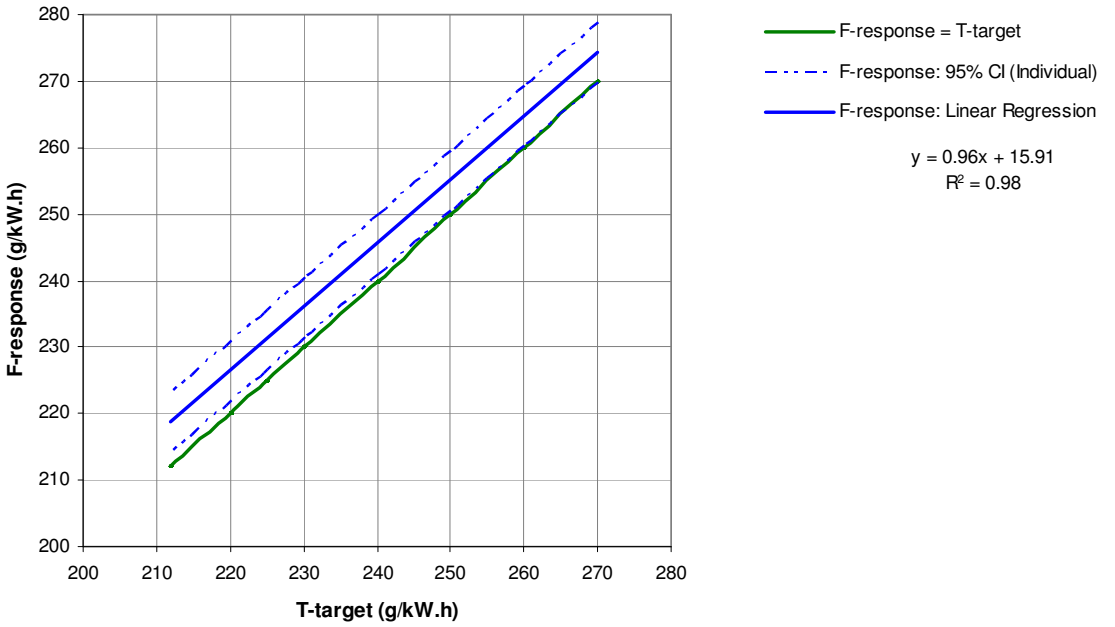


Figure 6.9(a): F-response regression to T-target response

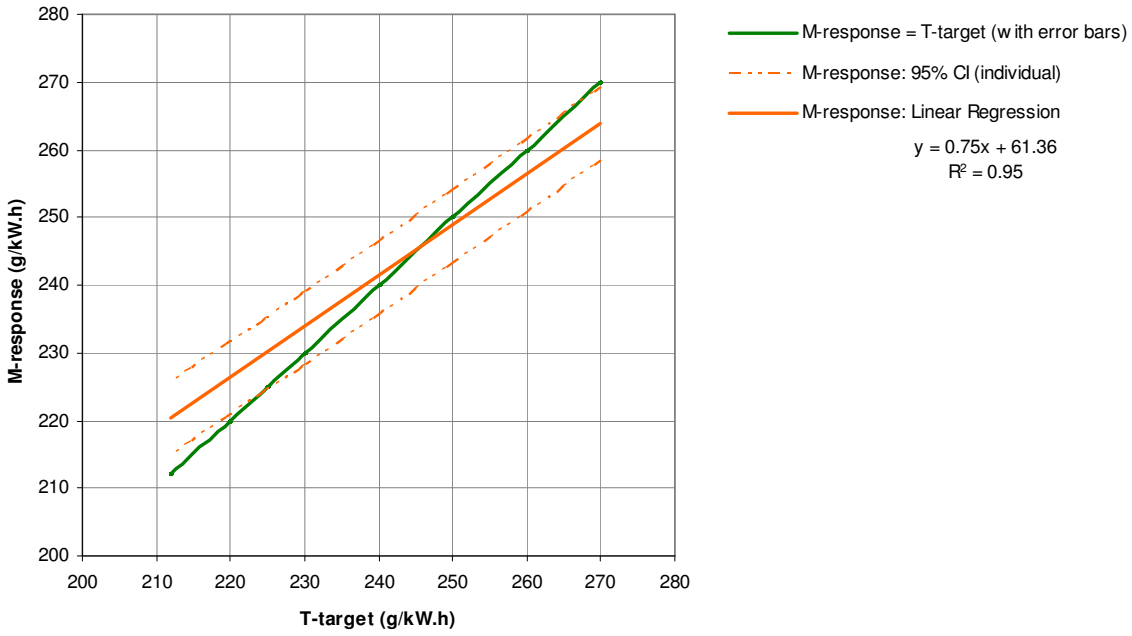


Figure 6.9(b): M-response regression to T-target response

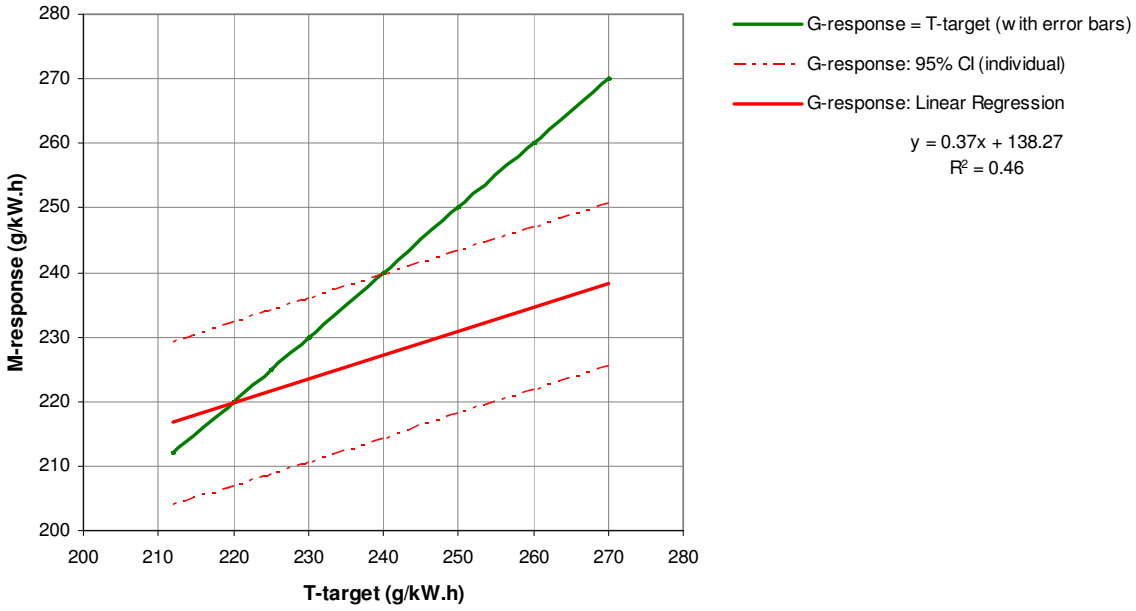


Figure 6.9(c): G-response regression to T-target response

Table 6.17 shows the regression model parameters for all three models as well as the goodness-of-fit parameters.

Table 6.17: Regression & Goodness of Fit Parameters – All Models

	Acceptance Limits	Model Response		
		F-response	M-response	G-response
Y Intercept	b_0	15.91 *(13.38-18.44)	61.36 (58.34-64.39)	138.27 (131.17-145.36)
Gradient	$b_1 \geq 0.75$	0.957 (0.947-0.968)	0.750 (0.738-0.763)	0.371 (0.341-0.400)
Coefficient of Efficiency	$\varepsilon_E \geq 0.85$	0.85	0.89	0.00
Modified Coefficient of Efficiency	$\varepsilon_{E_1} \geq 0.60$	0.57	0.66	-0.03
Index of Agreement	$\varepsilon_{IA} \geq 0.95$	0.96	0.96	0.65
Modified Index of Agreement	$\varepsilon_{IA_1} \geq 0.78$	0.78	0.81	0.45
Coefficient of Correlation	r	0.99	0.98	0.68
Coefficient of Determination	R^2	0.98	0.95	0.46

*(95% confidence interval of the prediction)

6.5.2 Goodness-of-Fit: Discussion on Findings

6.5.2.1 F-model Response to Target Response

With reference to Figure 6.9(a), the F-response to T-target regression line is higher than the perfect fit across all the contours. This can be seen by comparing the means of the F-response and T-target in Table 6.12 and Figure 6.5. Although the F-model response is generally higher than the target, it is generally parallel to the target line with the gradient b_1 at 0.957 (Table 6.17). Thus, indicating in general that the shape of the response surface is similar to that of the target surface when plotted on three dimensional coordinates (engine or mean piston speed, mean effective pressure and specific fuel consumption).

The F-model is validated on all the criteria except for the modified coefficient of efficiency. This indicates that, overall, the absolute difference between the model and target response was greater than the absolute difference between the target response and its mean response. This is clear from Figure 6.9(a).

6.5.2.2 M-model Response to Target Response

With reference to Figure 6.9(b), the M-response to T-target regression line is higher than the perfect fit line across the lower contours (212 and 220 g/kW.h), generally on target for the mid range contours (230, 240, 250 and 260 g/kW.h) and higher than target for the 270 g/kW.h contour. It is clear that the M-model has a good prediction performance on the test data mean (see Table 6.12). Although the M-model response has tracked the target mean response, it has been at the expense of the lower and upper specific fuel consumption portions of the performance map. With the gradient b_1 = 0.750 (Table 6.17), the general shape of the M-response surface will generally be flatter than that of the target when plotted on three dimensional coordinates (engine or mean piston speed, mean effective pressure and specific fuel consumption).

With respect to the goodness-of-fit parameters, the M-model is validated on all the parameters as shown in Table 6.17. Although the gradient is validated, the lower confidence level of the gradient falls below the acceptance limit and therefore should strictly not be accepted for validation.

6.5.2.3 G-model Response to Target Response

With reference to Figure 6.9(c), the G-response regression line is higher than the perfect fit line at the lowest contour (212 g/kW.h), on target for the next higher contour (230 g/kW.h) and significantly lower than target for the remaining higher valued contours. The indications are that, with the gradient of b_1 = 0.426 (Table 6.17), the general shape of the response surface would be a lot flatter than that of the target when plotted on three dimensional coordinates (engine or mean piston speed, mean effective pressure and specific fuel consumption). This effect can clearly be seen by comparing the surface plot of Figure 6.10(d) with that shown in Figure 6.10(a) – keeping note of the colour bar scale and the target contour line values.

The G-model response is not validated on any of the goodness-of-fit parameters. It is interesting to note that the coefficient of efficiency is zero, implying that the target response mean $\overline{Y}_R$ (or observed mean) is a better predictor than the G-model on the test data set.

Findings:

The results of acceptance and non acceptance of the alternative hypothesis for each regression and goodness-of-fit parameter are as follows:

Model Validation Acceptance Hypothesis	F-response	M-response	G-response
$H_a: \ \mathbf{y}_M \approx \mathbf{y}_R \quad \text{if} \quad b_1 \geq 0.75$	Yes	Yes	No
$H_a: \ \mathbf{y}_M \approx \mathbf{y}_R \quad \text{if} \quad \varepsilon_E \geq 0.85$	Yes	Yes	No
$H_a: \ \mathbf{y}_M \approx \mathbf{y}_R \quad \text{if} \quad \varepsilon_{E_1} \geq 0.60$	No	Yes	No
$H_a: \ \mathbf{y}_M \approx \mathbf{y}_R \quad \text{if} \quad \varepsilon_{IA} \geq 0.95$	Yes	Yes	No
$H_a: \ \mathbf{y}_M \approx \mathbf{y}_R \quad \text{if} \quad \varepsilon_{IA_1} \geq 0.78$	Yes	Yes	No

The M-model has demonstrated good fit to the test data and the validation is accepted for this model. The fit for the F-model is fair and validation has only not been accepted based on the modified coefficient of efficiency. The fit for the G-model is extremely poor and the model is not validated by the tests.

Thus far, the models have been tested against the training and validation data, their sensitivity and response behaviour checked against known diesel engine and thermodynamic theory, and the model response means tested against the target mean using the test data. Further, the model response to target correlation has been checked. The next step in the model validation and testing process is to review the model response across the entire performance map surface, as well as horizontal and vertical slices through the map.

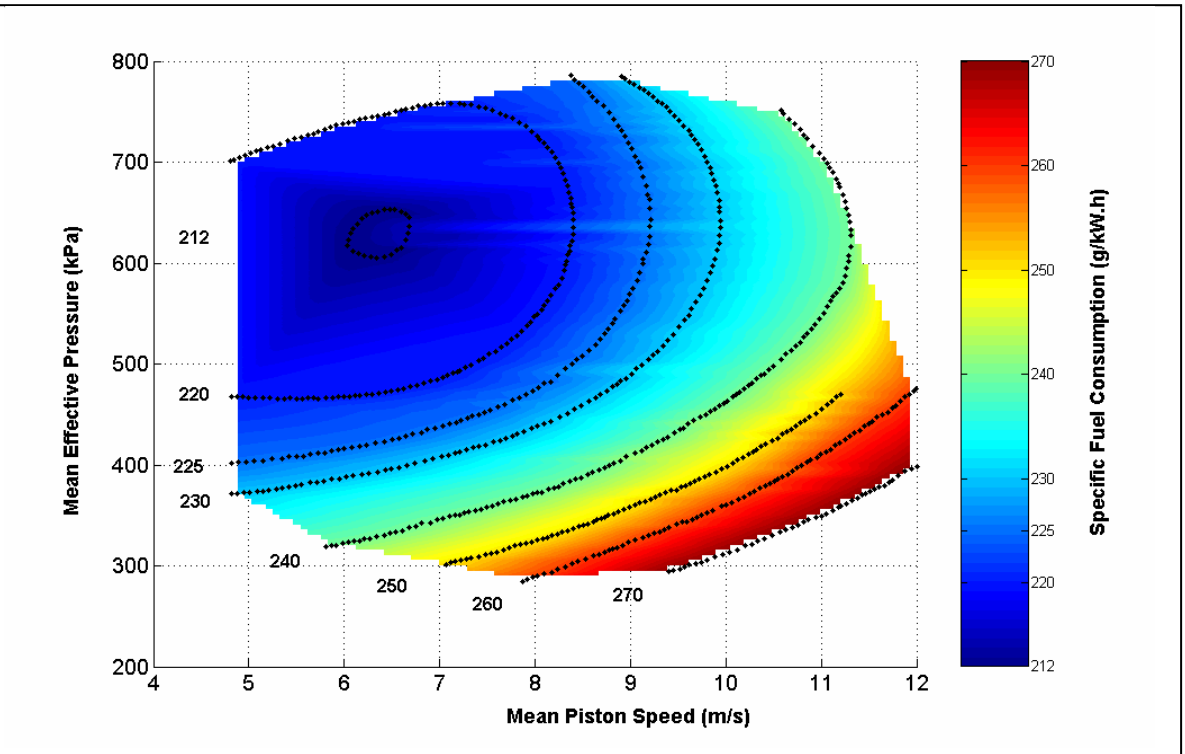
6.6 Response Surface Error: Interpolated Test Data

In this section the model response is reviewed over a horizontal and vertical grid over the surface of the test engine performance map. The review and tests performed here are as per the methodology defined in section 5.4.

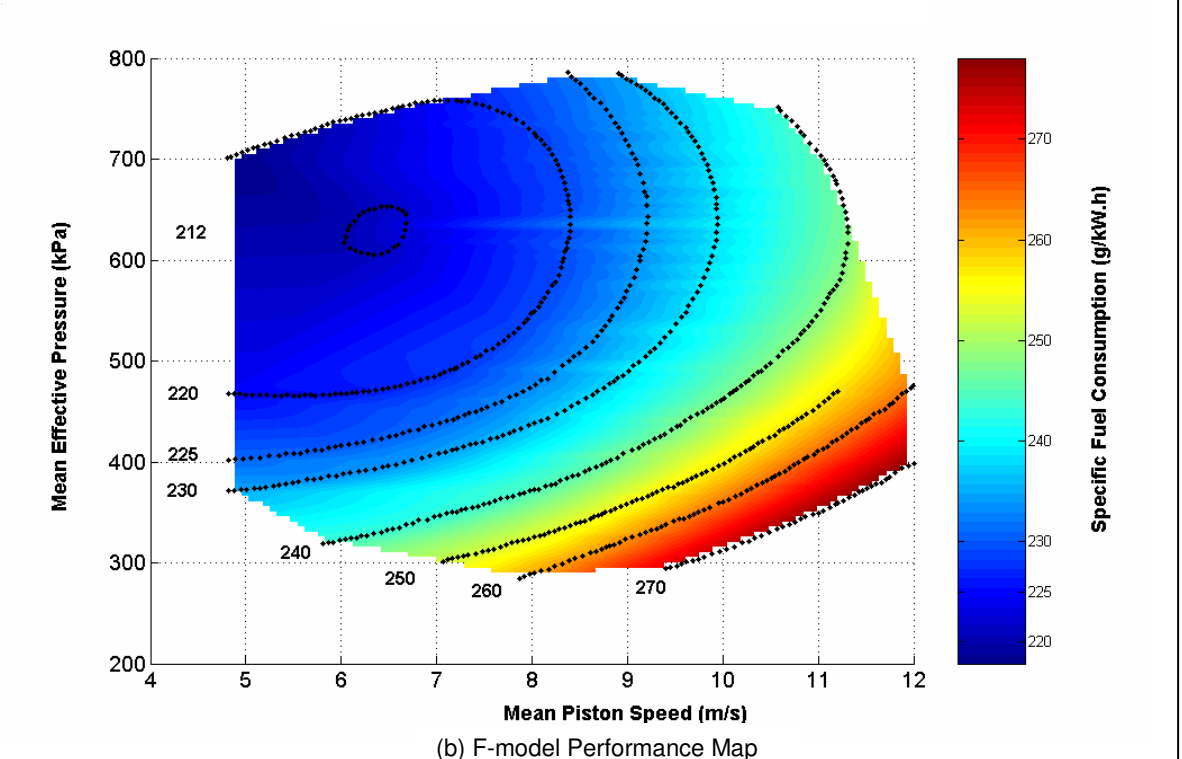
6.6.1 Model Response: Surface Maps on Interpolated Test Data

Figure 6.10(a) shows the interpolated test engine performance map on mean piston speed and mean effective pressure coordinates. As discussed in section 2.2.4, these coordinates allow for comparison of different engine performance maps that differ in both scale and shape. The labelled dotted lines on the plots are the constant specific fuel consumption contours (of the test data set) and the colour bar on the right indicates the specific fuel consumption value of the performance map surface.

It is clear from Figures 6.10(b) and 6.10(c) that the F-model and M-model response surfaces closely resemble that of the target (Figure 6.10(a)) in terms of general shape and scale. It is seen that the region of minimum specific fuel consumption is within the range 5 to 6 $m.s^{-1}$ which is a general characteristic of diesel engines (Taylor, 1985). For the G-model, the response surface shape is not similar to that of the target as can be deduced by examining Figure 6.10(c) in comparison with Figure 6.10(a). The G-model response surface is more symmetrical about its minimum specific fuel consumption; which is expected since the response is generated from a quadratic function. Further, the region of minimum specific fuel consumption (7.5 to 8.5 $m.s^{-1}$) is also in the incorrect position as expected from known diesel engine characteristics.

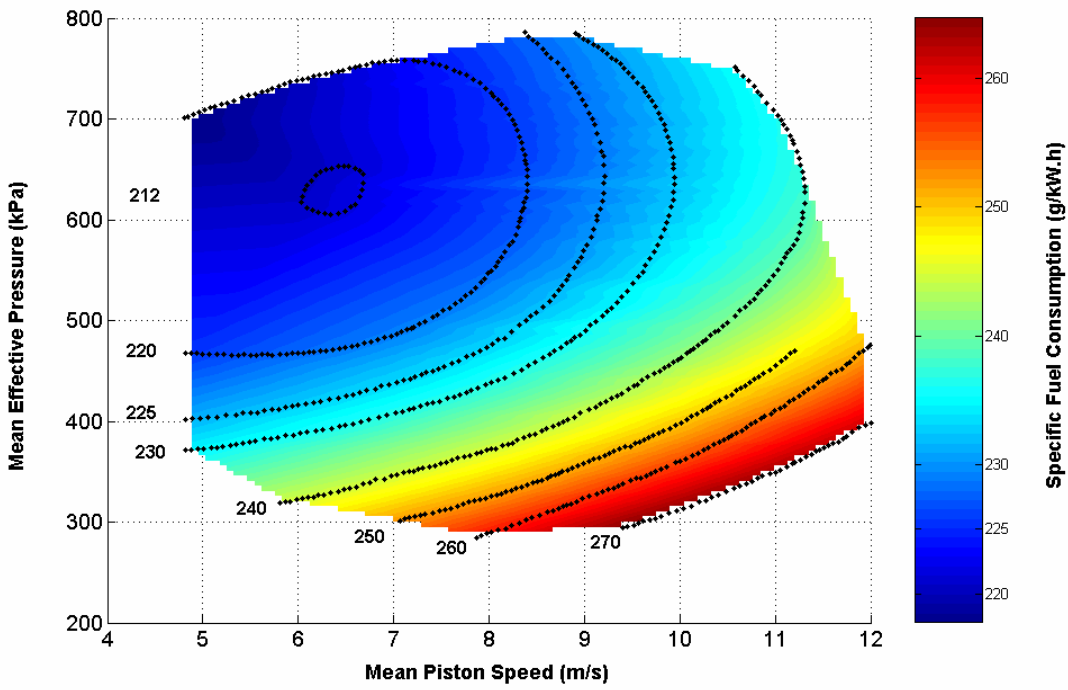


(a) Estimated Test Engine Performance Map

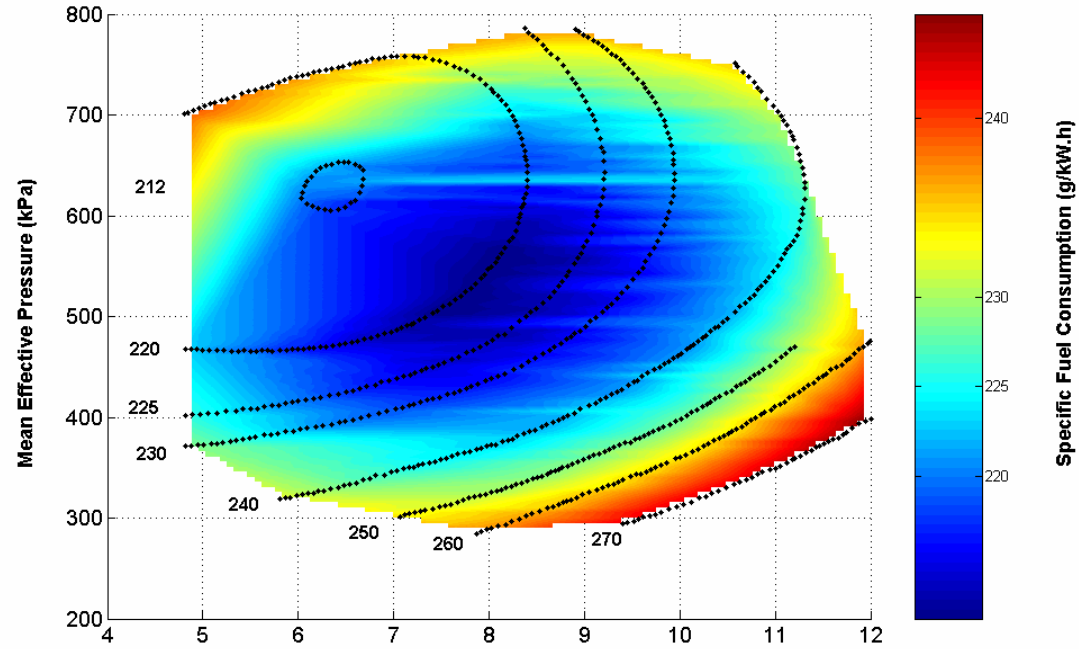


(b) F-model Performance Map

Figure 6.10: Engine Performance Map Surface Plot



(c) M-model Performance Map



(d) G-model Performance Map

Figure 6.10 (cont): Engine Performance Map Surface Plot

Although Figures 5.10(b) to 5.10(d) give the general shape of the model performance map, it is informative to check the difference between the model response and target response; reported as a percentage error for comparison with the allowable error.

6.6.2 Model Response Absolute Error Surface Plots (all models)

Here the difference in model response to target response is plotted as an error surface for the three models. The absolute error has been defined in section 5.4.1.1. Figures 6.11(a) to 6.11(c) show the absolute percentage error between the model response and the interpolated target response for all three models. The constant specific fuel consumption lines (or contours) are shown as dotted lines together with their values.

6.6.2.1 Model Validation: Performance Map Surface

Each model is validated across the surface of the map if the alternative hypothesis criterion is accepted as follows – see section 5.4.1.1:

$$H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{RI}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR}} \leq 1$$

The mean square error and the normalised mean square error have been calculated for the interpolated test data. For each model, the results of the calculations are shown in Table 6.18.

Table 6.18: Model MSE and MSER on Test Engine Performance Map

	F-model	M-model	G-model
MSE	44.9	30.2	51.4
MSER ($\varepsilon_{\mathrm{MSR}}$)	1.4	0.9	1.4
N = 1964 interpolated target data points			

F-model Results:

Figure 6.11(a) relates to the F-model error, and is the absolute error between the surface maps of Figure 6.10(a) and Figure 6.10(b). It is seen that the absolute error in prediction of the map using the F-model has an error in the range -0.5% to +4.5%. The error is not symmetrical and there is maximum error in a horizontal strip between

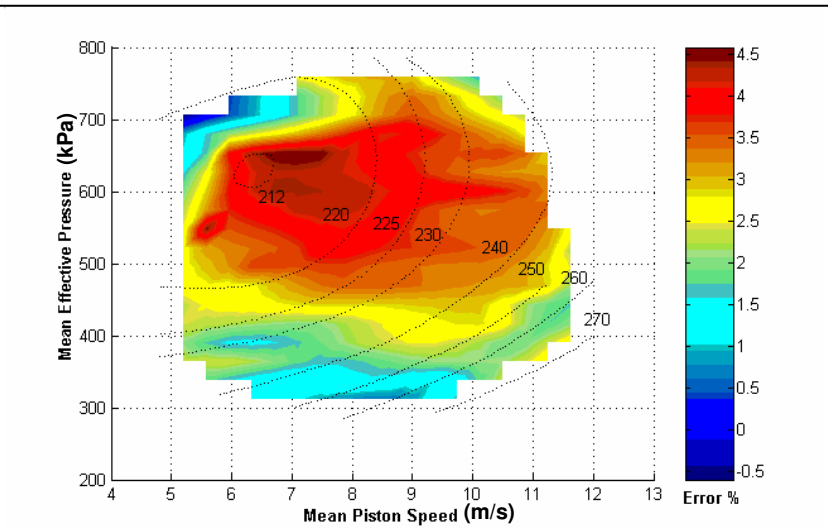
450 kPa and 650 kPa. The error is greater than the acceptable error of 2.5% over a large portion of the engine performance map. Further, from Table 6.18, it is clear that the F-model does not meet the validation criteria with a normalised mean squared error greater than 1. Therefore the alternative hypothesis for this model is not accepted.

M-model Results:

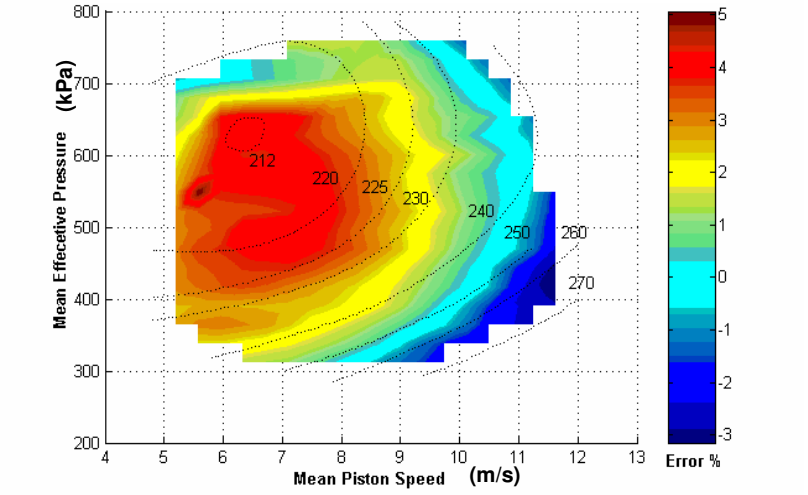
Figure 6.11(b) relates to the M-model absolute error and is the error between the surface maps of Figure 6.10(a) and Figure 6.10(c). The absolute error in prediction of the M-model has an error in the range -2.5% to +4.5%. The error is symmetrical and there is maximum error in the region of minimum specific fuel consumption. This symmetry may be advantageous in practical applications in that a response offset can be applied to the model output to compensate for the error. A notable finding is that the error is relatively low (-1% to 2%) at the mid range of the specific fuel consumption (230 to 260 g/kW.h) where many engines will typically be operated, that is, 80% speed and 20 to 80% power. The finding is that one can expect a lower total error in consumption prediction from the M-model in actual applications. This is supported by the earlier findings on the analysis of the overall mean. From Table 6.18 it is seen that the M-model response normalised mean squared error meets the validation criteria. Therefore the alternative hypothesis is accepted.

G-model Results:

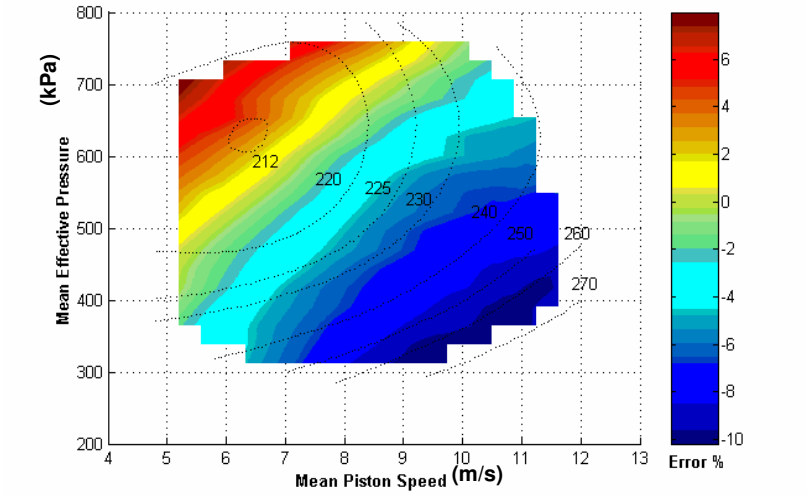
Figure 6.11(c) relates to the G-model performance and is the absolute error between the surface maps of Figure 6.10(a) and Figure 6.10(d). The absolute error in prediction of the G-model has an error in the range -10% to +6.5%. The error is not symmetrical and there is maximum error in the region of maximum specific fuel consumption. The error is high (-4% to -10%) at the mid range of the specific fuel consumption (230 to 260 g/kW.h) where many engines typically are operated. The finding is that one can expect a large total error in consumption prediction from the G-model. Table 6.18 shows that the validation acceptance criterion is not met for the G-model. However, it is notable that the G-model's error performance is similar to that of the F-model; even though it is a much simpler model and has not been trained using the training and cross-validation data sets.



(a) F-response Absolute Error on Interpolated Target



(b) M-response Absolute Error on Interpolated Target



(c) G-response Absolute Error on Interpolated Target

Figure 6.11: Model Response Absolute Error on Interpolated Target Response

Findings: $H_a: \quad \mathbf{y}_{\mathrm{M}} \approx \mathbf{y}_{\mathrm{RI}} \quad \text{if} \quad \varepsilon_{\mathrm{MSR}} \leq 1$ is only accepted for the M-model response.

In the section that follows, the model error performance on slices across the surface of the test engine performance map are reviewed.

6.6.3 Model Validation: Interpolated Test Data Map Slices

Horizontal (constant mean effective pressure) and vertical (constant engine speed) slices have been taken through the interpolated test engine performance map. The model response has been compared to the target for these slices. This has been done so as to get an improved view of the model predictive performance when the model would be applied in specific engine applications. The validation tests are performed as per the methodology of section 5.4.1.2.

6.6.3.1 Model Validation: Slices Through the Performance Map Surface

Each model is validated for the mean of the error for the horizontal and vertical slices as follows:

Horizontal or constant mean effective pressure slices:

$$H_a: \quad \mathbf{y}_{\mathrm{M_H}} \approx \mathbf{y}_{\mathrm{RI_H}} \quad \text{if} \quad \bar{\varepsilon}_{\mathrm{MSR_H}} \leq 1$$

Vertical or constant engine speed slices:

$$H_a: \quad \mathbf{y}_{\mathrm{M_V}} \approx \mathbf{y}_{\mathrm{RI_V}} \quad \text{if} \quad \bar{\varepsilon}_{\mathrm{MSR_V}} \leq 1$$

where $\bar{\varepsilon}_{\mathrm{MSR_H}}$ and $\bar{\varepsilon}_{\mathrm{MSR_V}}$ are the mean of the normalised mean squared error for the horizontal slices and vertical slices respectively and based on the nearer specification limit with $\varepsilon_{\mathrm{a}} = 0.025$.

6.6.3.2 Constant Mean Effective Pressure (Horizontal) Slices

Figures 6.12(a) to 6.12(e) show the model response at constant brake mean effective slices across the test engine performance map of Figure A.2.4 for the interpolated test data set. Here the plot scales are engine speed and specific fuel consumption, not mean piston speed and specific fuel consumption. Therefore the slices are taken through Figure A.2.4 and not Figure 6.10(a). The results are as follows:

F-model:

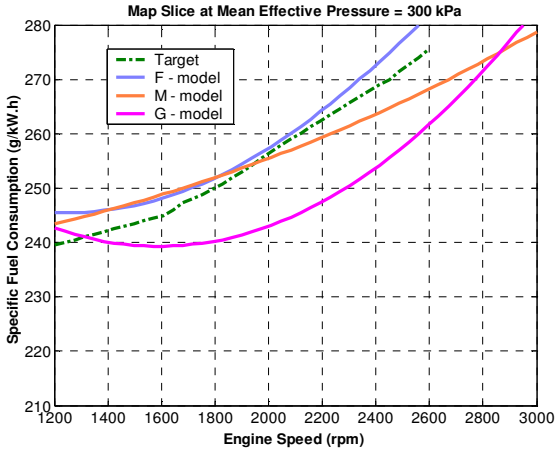
In general the F-model response follows the shape of the test response fairly well, with the specific fuel consumption being consistently higher than the target. That is, the response surface shape is correct at constant mean effective pressure but not the value. This was also found in sections 6.4 and 6.5. As seen in the error table, Table 6.19 of Figure 6.12, the mean squared error found for the F-model response is good at the lower mean effective pressure slices (300 to 400 kPa), however is relatively poor at the middle to upper slices (500 to 700 kPa). In the context of this performance validation review, mean squared errors below 35 $(g/kW.h)^2$ are considered acceptable. Table 6.19 shows that the average normalised mean squared error is greater than 1 for the F-model and the model validation on the alternative hypothesis is not accepted.

Findings: $H_a: \mathbf{y}_{M_H} \approx \mathbf{y}_{RI_H}$ if $\overline{\varepsilon}_{MSR_H} \leq 1$ *is not accepted for the F-model response on the horizontal slices.*

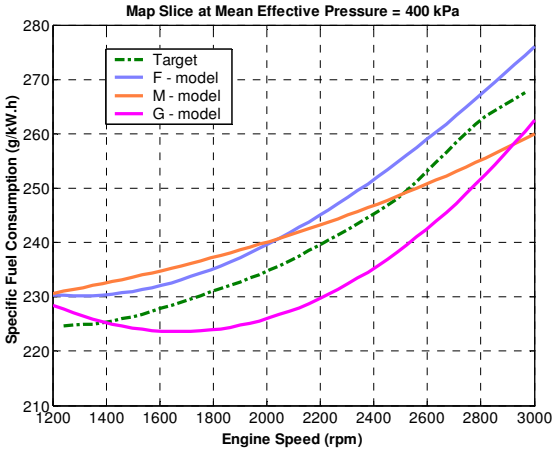
M-model:

The shape of the M-model response is in general flatter than that of the target with values following the target well in the middle to higher engine speeds. The error table, Table 6.19 of Figure 6.12, shows that the M-model response has a good error at the lower and higher mean effective pressure slices (300 to 400 kPa and 600 to 700 kPa) and a poor error on the mid-range slice (500 kPa). The mean of the normalised mean squared error was less than 1, therefore the model is accepted on the alternative hypothesis. However, the model response is close to the limit of the nearer specification limit.

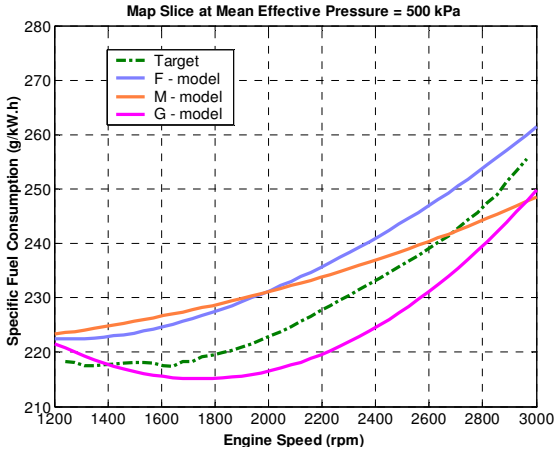
Findings: $H_a: \mathbf{y}_{M_H} \approx \mathbf{y}_{RI_H}$ if $\overline{\varepsilon}_{MSR_H} \leq 1$ *is accepted for the M-model response on the horizontal slices.*



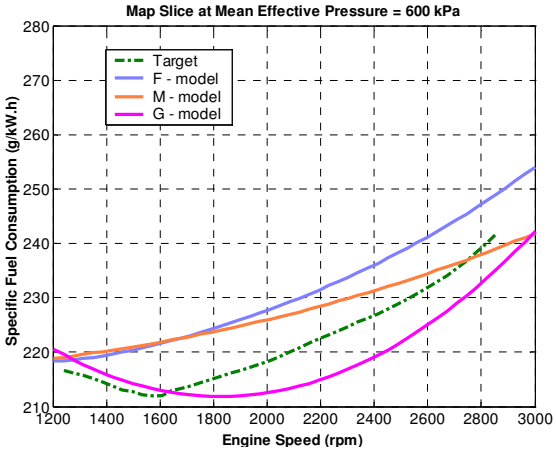
(a) Response at P_b = 300 kPa



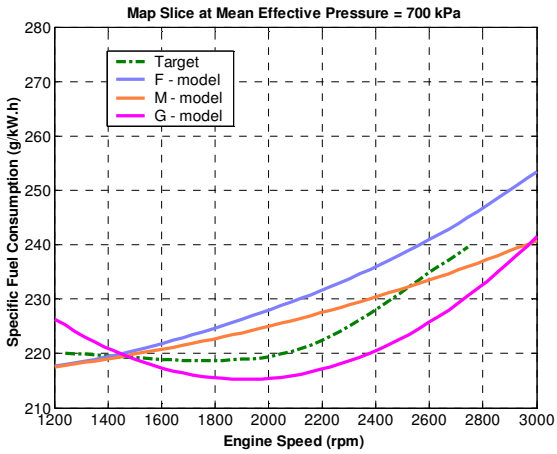
(b) Response at P_b = 400 kPa



(c) Response at P_b = 500 kPa



(d) Response at P_b = 600 kPa



(e) Response at P_b = 700 kPa

Table 6.19: Model Error at Constant Mean Effective Pressure

P_b (kPa)	F-model		M-model		G-model	
	MSE	MSER	MSE	MSER	MSE	MSER
300	12.0	0.3	14.5	0.3	122.9	2.9
400	27.8	0.8	35.5	0.9	67.1	1.8
500	52.6	1.6	42.5	1.4	38.8	1.1
600	72.2	2.3	39.4	1.3	30.4	1.0
700	40.4	1.3	11.7	0.4	30.5	0.9
Mean	**41.0**	**1.3**	**28.1**	**0.9**	**57.9**	**1.5**

Figure 6.12: Model Response at Constant Brake Mean Effective Pressure Slices

G-model:

In Figure 6.12, in general, the shape of the G-model response follows that of the target, with the shape being more concave and the values being lower. The error table, Table 6.19 of Figure 6.12, indicates that the model response is good in the mid-range and higher slices (500 to 700 kPa) but is very poor in the low range slices (300 to 400 kPa) for the G-model response. With a mean normalised mean squared error greater than 1, the model is not validated on the alternative hypothesis. Again, it is interesting to note that the error performance of the G-model is similar to that of the F-model response error.

Findings: $H_a: \mathbf{y}_{M_H} \approx \mathbf{y}_{RI_H}$ if $\bar{\varepsilon}_{MSR_H} \leq 1$ *is not accepted for the G-model response on the horizontal slices.*

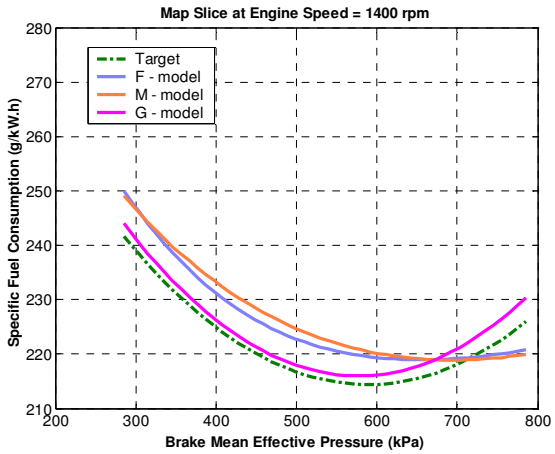
6.6.3.3 Constant Engine Speed (Vertical) Slices

Similar to the previous section, Figures 6.13(a) to 6.13(h) show the model response to target response at constant engine speed slices through the test engine performance map of Figure A.2.4. The following can be observed from Figures 6.13(a) to 6.13(h):

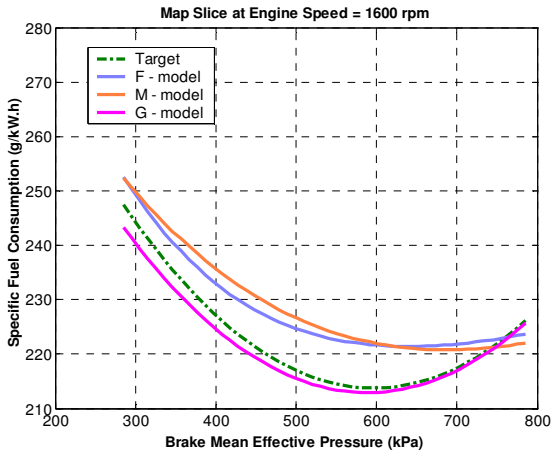
F-model:

The values are in general higher than the target with the shape, in general, only following the target in the middle to upper speed range. The curvature of the model response is a lot less than that of the target at the higher mean effective pressure values. This is especially noticeable at the lower and higher engine speed slices. The normalised mean square errors (MSER) for each speed slice are shown in Table 6.20 of Figure 6.13. It is seen that MSER for each slice is outside the nearer specification limit, correspondingly, the mean of errors is greater than 1 and the alternative hypothesis is not accepted for the model validation.

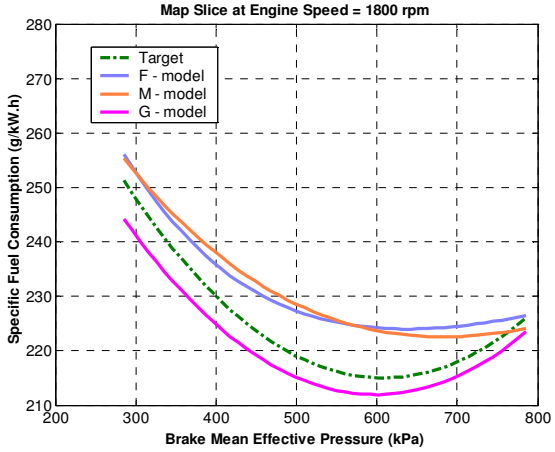
Findings: $H_a: \mathbf{y}_{M_V} \approx \mathbf{y}_{RI_V}$ if $\bar{\varepsilon}_{MSR_V} \leq 1$ *is not accepted for the F-model response on the vertical slices..*



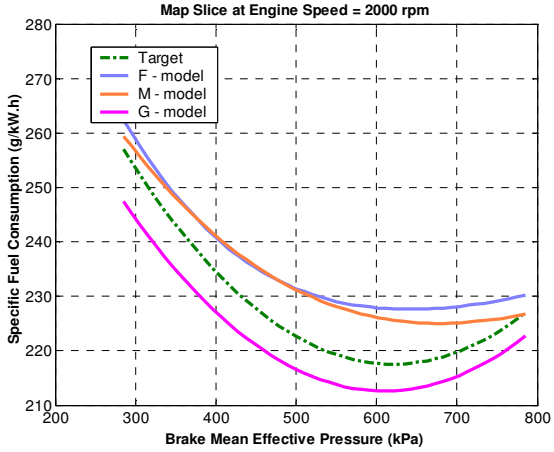
(a) Response at N = 1400 rpm



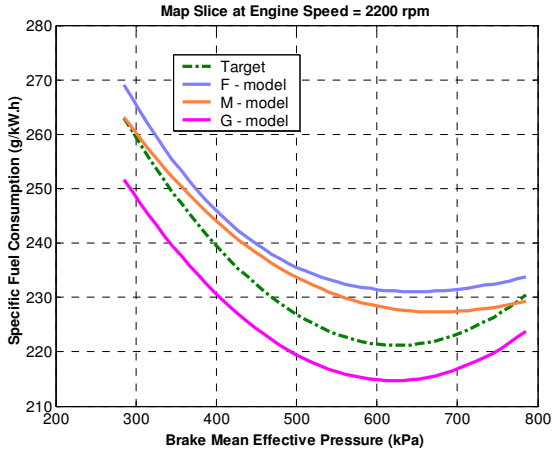
(b) Response at N = 1600 rpm



(c) Response at N = 1800 rpm

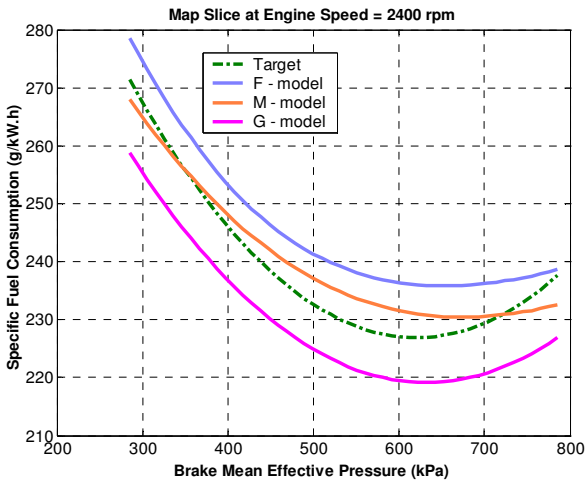


(d) Response at N = 2000 rpm

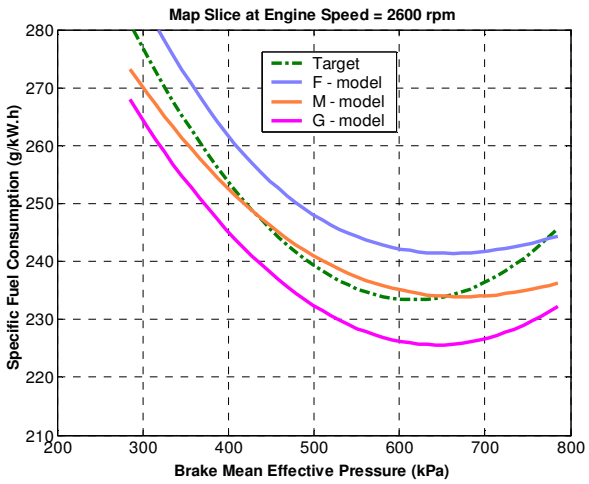


(e) Response at N = 2200 rpm

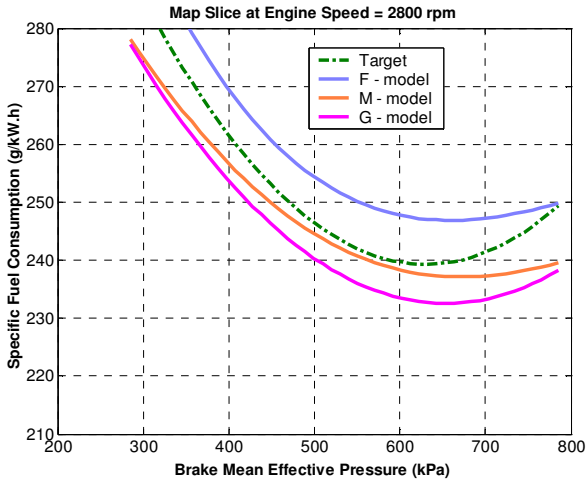
Figure 6.13: Model Response at Constant Engine Speed Slices



(f) Response at N = 2400 rpm



(g) Response at N = 2600 rpm



(h) Response at N = 2800 rpm

Table 6.20: Model Error at Constant Engine Speed

N (rpm)	F-model		M-model		G-model	
	MSE	MSER	MSE	MSER	MSE	MSER
1400	29.3	0.9	42.7	1.3	4.8	0.2
1600	37.5	1.2	53.2	1.7	3.9	0.1
1800	47.8	1.6	52.7	1.7	18.0	0.5
2000	61.4	2.0	43.8	1.4	39.8	1.2
2200	61.8	1.9	26.4	0.8	62.1	1.8
2400	58.2	1.7	11.0	0.3	82.5	2.3
2600	56.4	1.5	13.5	0.3	86.6	2.2
2800	55.6	1.4	28.6	0.6	66.1	1.6
Mean	**81.6**	**1.5**	**34.0**	**1.0**	**45.5**	**1.2**

Figure 6.13 (cont): Model Response at Constant Engine Speed Slices

M-model:

The M-model response is similar to the F-model response at the lower speed slices, however is nearer that of the target for the middle to higher engine speed slices. The curvature of the response is flatter than the target at the higher mean effective pressure values for each engine speed slice. The model's error performance is best in the middle to higher engine speed slices. Table 6.20 indicates that the M-model meets the mean MSER criteria, and therefore the model is validated on the alternative hypothesis. Again, as for the constant mean effective pressure slices, the model response error is on the upper limit of the nearer specification limit.

Findings: $H_a: \mathbf{y}_{\mathrm{M_V}} \approx \mathbf{y}_{\mathrm{RI_V}}$ if $\bar{\varepsilon}_{\mathrm{MSR_V}} \leq 1$ *is accepted for the M-model response on the vertical slices.*

G-model:

The G-model response follows the general shape of the target response very well with values generally lower than those for the target. In terms of general shape, the G-model outperforms that on the NN models. Study of Table 5.20 shows that the mean of the MSER for the G-model is greater than 1, therefore the model is not validated on the alternative hypothesis. However, the G-model response error outperformed that for the F-model. That is, the G-model seems a better predictor than the F-model of the specific fuel consumption at constant speed and variable mean effective pressure.

Results summary: $H_a: \mathbf{y}_{\mathrm{M_V}} \approx \mathbf{y}_{\mathrm{RI_V}}$ if $\bar{\varepsilon}_{\mathrm{MSR_V}} \leq 1$ *is not accepted for the G-model response on the vertical slices.*

6.7 Summary of Findings

In summary, analysis of the model performances yielded the following results:

- The model performance verification and validation is summarised in Table 6.21 (note the legend at the end of the table for reference).

- There were a total of twenty three verification and validation tests performed on the neural network models (excluding the four significance tests). A total of eleven validation tests were performed on the quadratic model.

- Passing scores?

- For the F-model, the verification and validation alternative hypothesis was accepted fifteen out of twenty three times (or a score of 65%). That for the M-model was accepted twenty two times out of twenty three (or a

score of 96%). For the G-model, the alternative hypothesis was not accepted for all eleven tests (or a score of zero).

- The F-model was not validated on the horizontal and vertical interpolated test data. This finding means that the F-model should not be used in applications where constant speed and variable torque or constant torque and variable speed are the mode of engine operation.

- The G-model is clearly invalid for the application. It had very poor performance in terms of the verification and validation tests. However, in some instances it outperformed the F-model. This is noteworthy, since the G-model has a much simpler construct. Considering its simplicity, the G-model demonstrated reasonable performance.

Both neural network models outperformed the quadratic method in terms of model verification and validation for the application. In terms of the alternative hypothesis acceptance score, the M-model (modular network) can be taken as better validated for the applicant over the F-model (non-modular network).

When comparing the validation performance of the models, strictly one cannot use the test data set to decide on the better validated model. If the test data set is used for alternative model validation performance comparison, it then by definition becomes part of the cross-validation data set. This then defeats the purpose of having an independent test data set. However, since no model parameter tuning (or new learning) was done during the tests and the verification and validation methodologies employed were deemed to be sufficiently critical, here the test data set has been used to select the best model for the application. In this case the modular neural network.

Since a neural network model has been validated, it is beneficial to demonstrate a typical application of the model in a formalised energy monitoring and targeting system.

Table 6.21: Verification and Validation – Summary of Verification and Validation Findings

Verification or Validation Test	Section Reference	Data Set	Verification	Validation	Type	F-model	M-model	G-model
1. NN Training and Weight Optimisation								
1.1. Maximum network weights criteria	6.2.1	Train	X		WB			
1.2 Network Training Error Goal:								
MSE	6.2.2	Train	X		WB			
MSE	6.2.2	CV	X		WB			
AIC	6.2.3	Train	X		WB			
AIC	6.2.3	CV	X		WB			
2. Difference in Network to Network Training Performance								
2.1. Individual NN Model:								
MSE (Training to Cross-validation data sets)	6.2.2.1	Train/CV	X		SIG			
AIC (Training to Cross-validation data sets)	6.2.3.1	Train/CV	X		SIG			
2.2. Across NN Models:								
MSE (Training to Cross-validation data sets)	6.2.2.2	Train/CV	X		SIG			
AIC (Training to Cross-validation data sets)	6.2.3.2	Train/CV	X		SIG			
3. Sensitivity Analysis								
3.1. Raw Sensitivity	6.3.1	Train	X		WB			
3.2. Input-Output Mapping								
Mean Piston Speed	6.3.2.1	Train		X	WB			
Brake Mean Effective Pressure	6.3.2.2	Train		X	WB			
Specific Power	6.3.2.3	Train		X	WB			
Piston Area to Volume Displaced	6.3.2.4	Train		X	WB			
Standard Rated Specific Power	6.3.2.5	Train		X	WB			
Standard Rated Specific Fuel Consumption	6.3.2.6	Train		X	WB			
4. Predicted Mean								
4.1. Overall Mean (within NSL)								
Confidence Level Limits Test	6.4.1.1	Test		X	BB			
Normalised Mean Square Error Limits Test	6.4.1.2	Test		X	BB			
4.2. Contour Means								
Model response mean at target contour mean	6.4.2	Test		X	BB			

Verification or Validation Test	Section Reference	Data Set	Verification	Validation	Type	F-model	M-model	G-model
5. Goodness-of-Fit: Model to Target Response								
Regression gradient	6.5.1	Test		X	BB			
Coefficient of efficiency	6.5.1	Test		X	BB			
Modified coefficient of efficiency	6.5.1	Test		X	BB			
Index of agreement	6.5.1	Test		X	BB			
Modified index of agreement	6.5.1	Test		X	BB			
6. Error on Surface Maps								
Normalised Mean Square Error Limits Test	6.6.2.1	Int-test	X		BB			
Constant mean effective pressure slices (horizontal)	6.6.3.2	Int-test	X		BB			
Constant engine speed slices (vertical)	6.6.3.3	Int-test	X		BB			

Notes:

	H_0 not rejected (no difference in performance)	**Train**	Training set	**WB**	White-box validated
	H_0 rejected (a difference in performance)	**CV**	Cross-validation	**BB**	Black-box validated
	H_a accepted – model verified/validated	**Test**	Test	**SIG**	Significance (t test)
	H_a not accepted – model not verified/validated	**Int-test**	Interpolated Test		

Chapter 7

Model Application Demonstration

7.1 Introduction

In this chapter an application of the diesel engine model is demonstrated in terms of its typical use in an energy monitoring and targeting management system. The practical aspects of the model application and implementation processes are described together with the application limitations. This demonstration is based on a synthetic set of engine operational data and uses the validated modular neural network (M-model).

The formal process of energy monitoring and targeting (M&T) has been summarised in the introduction. Here, the statistical tools typically used in the process of M&T are highlighted and referenced. The key aspect of the demonstration is to show the critical importance of the engine performance mapping model in relation to the process of M&T.

7.2 Energy Monitoring and Targeting

As highlighted earlier, success of the M&T process depends on being able to recognise when there is a difference between actual consumption and the expected or target consumption for the same period. Further, is this difference exceptional? In the context of diesel engine performance M&T, the creation of the information to support understanding requires two essential data streams and an estimating tool (process model) for each energy consumption period:

- the actual fuel consumption or energy usage for the period;
- a measurement of the primary operational parameters so that the energy driver, in this case power generated, can be calculated for the period; and
- a tool or model to estimate the target or desired consumption for the period based on the energy drivers.

For any one period the engine monitoring is performed as follows:

- The actual fuel consumption is measured;
- The actual power generated driving the fuel consumption is calculated from the operating parameters and the target fuel consumption is estimated using the engine performance model and the actual operating parameters;
- The difference between the actual consumption and the target consumption - the variance in performance – can then be used to indicate whether there has been an exception in fuel consumption for the period.

The process is therefore quite simple and answers the question: “what was the actual consumption versus what should it have been?” The numerical difference between the actual consumption and “should have been” or target consumption is called the fuel consumption variance. Estimation of the fuel consumption variance is critical to the M&T process and is estimated as follows:

7.3 Calculation of Fuel Consumption Variance

The engine fuel consumption performance variance can be stated as follows:

$$\Delta M_{\mathrm{c}} = M_{\mathrm{ca}} - M_{\mathrm{ct}} \tag{7.1}$$

where M_{ca} and M_{ct} are the actual and target fuel mass consumption respectively, and are respectively calculated as follows:

7.3.1 Actual Fuel Consumption (M_{ca})

The actual total fuel mass consumption for time period t is given by

$$M_{ca} = \int_0^t \dot{m}_{ca}(t)dt \tag{7.2}$$

where the fuel mass flow rate $\dot{m}_{ca}$ (calculated from the volumetric flow rate as measured by a temperature and density compensated fuel meter), is given as:

$$\dot{m}_{ca} = \dot{v}_{ca}(\rho_{\mathrm{ref}} - \alpha T_{ca}) \tag{7.3}$$

where $\dot{v}_{ca}$ is the volumetric flow rate as measured by the fuel flow meter, ρ_{ref} is the fuel reference density, α is a temperature compensating coefficient and T_{ca} is the temperature of the fuel at the fuel meter.

7.3.2 Target Fuel Consumption (M_{ct})

The target fuel consumption in the time period t can be estimated by:

$$M_{\text{ct}} = \int_0^t \dot{m}_{\text{ct}}(t)dt = \frac{A_p}{4}\int_0^t Q_{\text{ct}}(t)\,.\,P_b(t)\,.\,\overline{U}_p(t)\,dt \tag{7.4}$$

where A_p is the piston area, P_b is the mean effective pressure, $\overline{U}_p$ is the mean piston speed, and Q_{ct} is the specific fuel consumption estimated using the neural network model for the engine (which is itself a function of the mean effective pressure and the mean piston speed).

The mean effective pressure and mean piston speed define the engine power and therefore the primary parameters driving fuel consumption. Mean effective pressure and mean piston speed are calculated using equation (2.11) and (2.14) respectively, and these equations are repeated here for clarity:

$$P_b = \frac{4\pi\tau}{V_d}$$

where the engine torque τ can be measured using a magnetostrictive torque sensor.

$$\overline{U}_p = 2Ns$$

where the engine speed N is measured using a tachometer.

It is clear that the fuel consumption variance cannot be estimated without the availability of a model that can estimate the target fuel consumption - $Q_{\text{ct}}(t)$. Although critical to the process, estimating the target fuel consumption is only part of the total solution. The M&T process still has to answer the questions around the validity of the variance. That is, has the engine performance actually changed or is the fuel consumption variance due to random events? This issue is effectively addressed by Change Point Analysis (CPA) developed by Taylor (2000) – details of the CPA procedure are detailed in Appendix A.3.5.

In the sections that follow, the use of the neural network model and the CPA method has been demonstrated using an example of an engine running on a simulated operating cycle.

7.4 Demonstration via Simulation

For this demonstration Engine No.4 (the test engine) has been used. The engine parameters for this engine are shown in Table A.1.1 and its performance map is shown in Figure A.2.4.

7.4.1 Simulated Duty Cycle

The simulated operating cycle is shown in Figure 7.1 (a) and (b). The engine power generated has been randomly varied each hour for 48 hours in the range 35 kW to 70 kW or 43% to 85% of the rated power of the engine. The varying engine power is shown by Figure 7.1(a). The engine speed has been varied in the range 1300 to 2600 revolutions per minute or 46% and 93% of the engine rated speed. The resulting engine brake torque is calculated and is shown together with the engine speed in Figure 7.1(b).

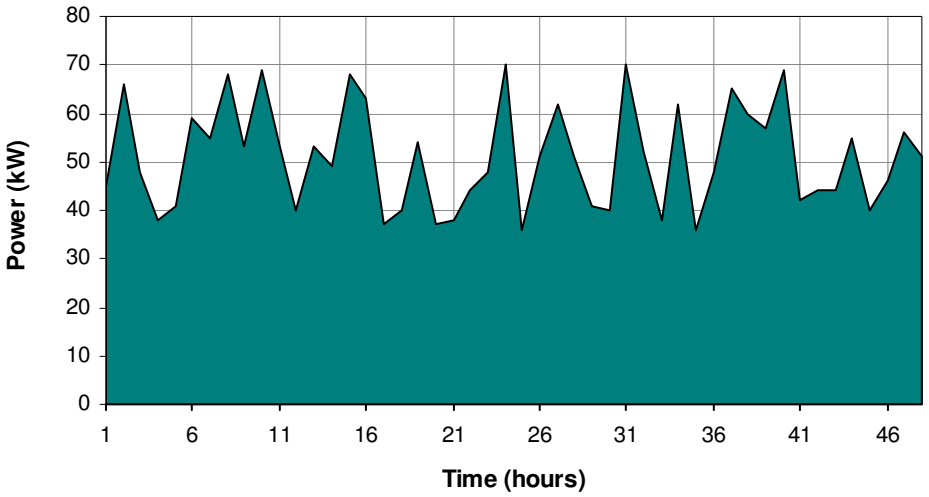
Using the engine parameters of Table A.1.1, and for the duty cycle of Figure 7.1(a) the engine rotational speed and brake torque have been converted into mean piston speed and mean effective pressure respectively. This has been done since mean piston speed and mean effective pressure are input variables to the neural network model.

7.4.2 Simulated Actual and Target Fuel Consumption

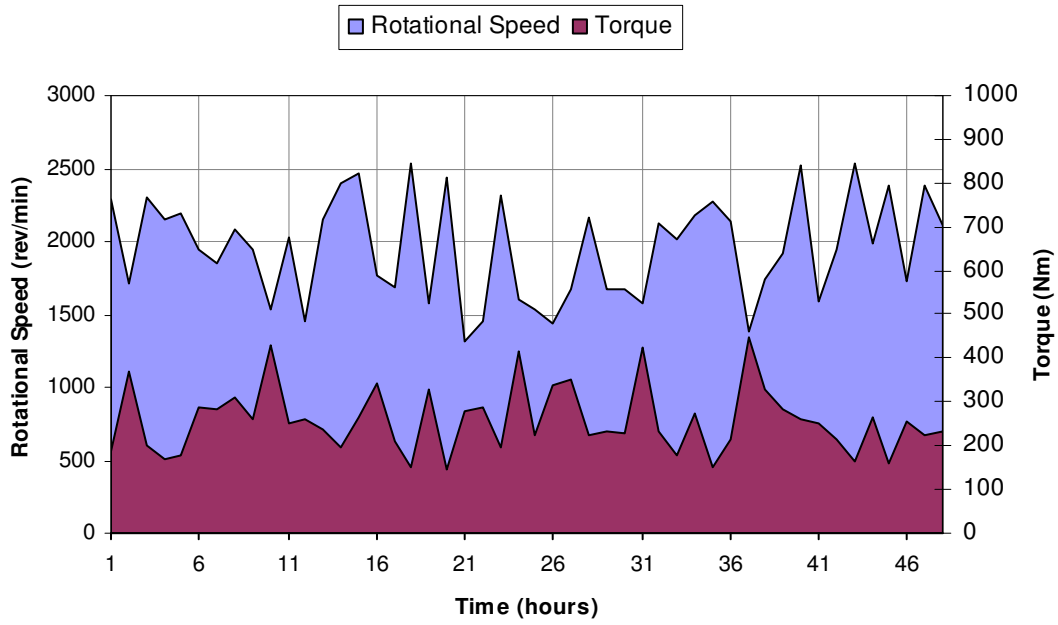
The target fuel consumption is estimated by using the modular neural network model to generate the specific fuel consumption for each hour in the duty cycle. This is then converted into consumption using equation (7.4).

Since no actual consumption exists for this demonstration, the actual fuel consumption is generated using the neural network model. This is calculated as for the target consumption. The actual fuel consumption has been simulated by adding

a random error to the model output. This error is based on the typical instrument propagated error. This is accomplished by allowing for a -2.5% to 2.5% random error around the calculated consumption using the model (see Appendix A.3.1 for the discussion on propagated error). Further, an engine fault that would increase the specific fuel consumption is simulated by randomly increasing the specific fuel consumption by 0% to 5%, starting at hour 32. That is, the CPA should detect a change or shift in the fuel consumption variance at or close to hour 32.



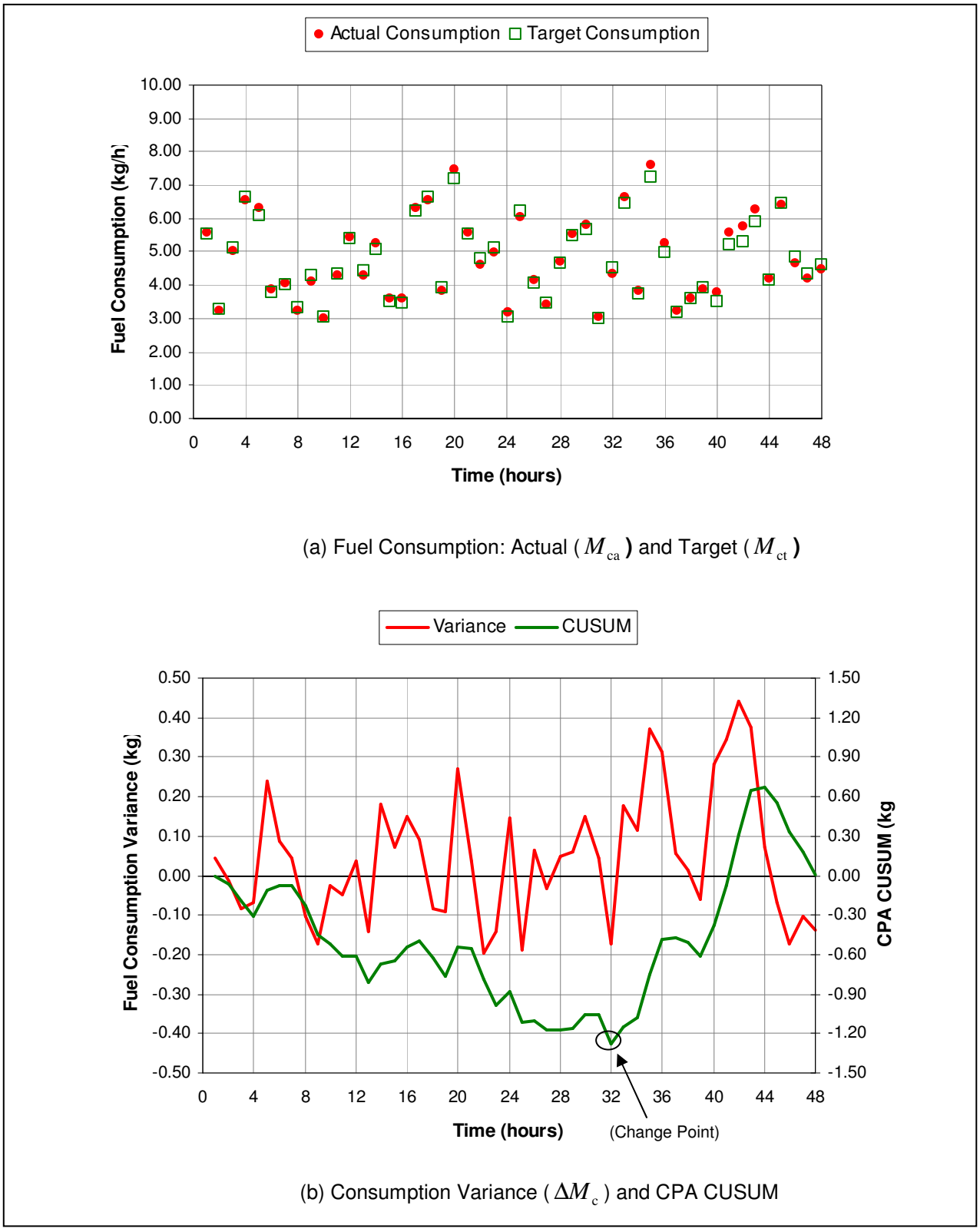
(a) Engine Power



(b) Engine Speed and Torque

Figure 7.1: Demonstration Engine Simulated Operating Cycle

The simulated actual and target fuel consumption for each operating hour for the duty cycle is shown in Figure 7.2(a) and the consumption variance (target to actual) is shown in Figure 7.2(b).



(a) Fuel Consumption: Actual (M_{ca}) and Target (M_{ct})

(b) Consumption Variance (ΔM_c) and CPA CUSUM

Figure 7.2: Demonstration Engine: Consumption and Variance (Actual and Target)

7.4.3 Findings

Considering Figure 7.2(a), the actual fuel consumption is plotted together with the target fuel consumption. Although typical in actual energy M&T systems, this form of graph is ineffective since it provides little information or insight into the change in fuel consumption. Figure 7.2(a) is presented here to demonstrate how small the difference is between the actual and target consumption. Many energy M&T systems also plot the consumption variance over time as shown in Figure 7.2(b) – left hand scale. Although providing a little more information than Figure 7.2(a). Since the variance seems to have increased in the second half of the period, this form of graph continues to hide relevant information. Extracting the hidden information is the reason why CPA is proposed for use when using the NN model for energy management.

Much improved insight can be obtained by plotting the CPA CUSUM as shown in Figure 7.2(b) – right hand scale. When interpreting the CUSUM, one must keep in mind that the CPA CUSUM is not the cumulative sum of the fuel consumption variance. Rather it is the cumulative sum of the difference between the variance and the mean of the variance for the period under review (in this case 48 hours). This approach, as proposed by Taylor (2000), tends to better graphically highlight the change point/s. The CPA CUSUM clearly shows that the consumption variance changed (or increased) somewhere close to hour 32 (note the change in a general negative gradient to a general positive gradient). A confidence level of the apparent change can now be estimated using the bootstrapping method detailed in Appendix 3.5.

Table 7.1 shows the results of the Change Point Analysis on the actual to target fuel consumption variance using a bootstrap of 1000 samples and a confidence interval of 95%.

Table 7.1: Demonstration CPA – Significant Change in Fuel Consumption Variance

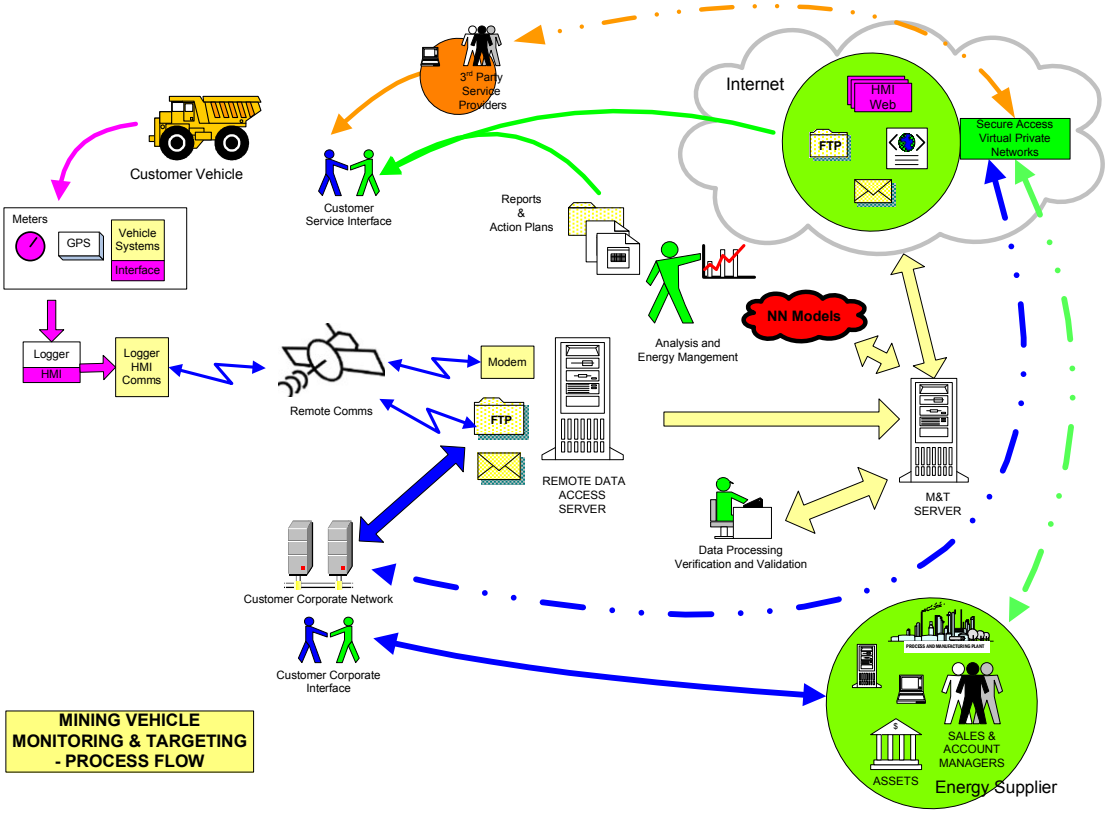
Time Point (hour)	Confidence Interval (hour to hour)	Confidence Level	Move From (kg/h)	Move To (kg/h)
34	(22 to 48)	98%	0.006125	0.12619

As shown in Table 7.1, the CPA found a change point with a 98% confidence level at time 34 hours, where the mean fuel consumption variance moved from 0.006125

kg/h to 0.12619 kg/h. This is one hour after the engine was simulated as having a problem that resulted in the specific fuel consumption increasing. Note that the mean fuel consumption variance should have been zero before the change point. This would have been the case if a larger data set had been simulated.

7.5 Actual System Implementation

As previously stated, the neural network model is only a component of a much larger



M&T system; albeit a critical component. The actual M&T systems and process would typically be as shown in Figure 7.3.

Figure 7.3: Remote Vehicle Monitoring & Targeting Systems and Process

This is an example of an actual implemented M&T system on large diesel engine powered mining vehicles. Major sections of the system are remote vehicle operational data acquisition, data verification and validation, database management and the reporting interfaces. Vehicle energy performance information is available near to real-time on Internet virtual private networks. Also, the technology is such that the HMI (Human Machine Interface) on the vehicle has imbedded performance models and reporting that is presented as a web page – locally and via the Internet.

This allows for real-time performance indication to the local operator and any remote person. Therefore, all the main groups involved in and responsible for the vehicle performance have direct access to critical performance information. These groups are typically the mine management team, the vehicle service team, the energy supplier team and the energy M&T team.

The result of the successful implementation of the M&T system together with the neural network based engine performance models, statistical methods, and communication technology is reduced energy consumption and pollutant emissions.

7.6 Discussion

It has been demonstrated, that using a model that can predict an engine's fuel consumption performance based on actual operational parameters in combination with an applicable statistical technique, results in a powerful component of a M&T management system. Such power allows for the effective understanding of the actual engine performance that ultimately leads to intervention and consumption or energy saving solutions over time.

The use of the neural network based engine performance model has some intrinsic limitations in its application on actual engines.

- The model is only of value if applied within a formalised monitoring and targeting system;
- The engine and its associated machine or plant needs to be fitted with the required operational parameter measurement instrumentation;
- It requires the acquisition of operational data that is often difficult or not cost effective to obtain.

Chapter 8

Conclusion

8.1 Introduction

This study was founded on the hypothesis that a neural network based diesel engine performance model can predict the specific fuel consumption performance of any specified normally-aspirated direct-injection diesel engine; within acceptable errors and practical application constraints. The development of such a model required that four primary sub-problems be addressed. These four primary sub-problems were:

1. Definition of the engine parameters that best describe diesel engine performance;

2. Selection and partitioning of engine performance data which adequately describe the problem;

3. Neural network selection, training and optimisation;

4. Neural network verification and validation, and comparison with the non-neural network model.

In conclusion, addressing the four primary sub-problems produced the following for a specified normally-aspirated direct-injection diesel engine within the parameter ranges defined by Table A.1.1:

- A set of geometric and operational parameters defining diesel engine specific fuel consumption performance; for use in a neural network model.

- A defined methodology for selection and partitioning of training and validation data for the neural network development.

- A defined methodology for the training of a neural network for diesel engine performance modelling, and specifically the application of a genetic algorithm for the optimisation of training parameters and neural network size.

- A defined set of test methodologies and acceptance criteria for the verification and validation of a neural network for the application of this study.

- A trained and validated neural network that has demonstrated performance improvement over a published quadratic modelling method.

- A demonstration of the implementation of the neural network in an energy monitoring and targeting system.

These are discussed and conclusions drawn in the sections that follow.

8.2 Engine Performance Parameters

A two-layer feed-forward network with sigmoid nonlinearity can approximate any function with arbitrary accuracy. Based on this fact, why bother expending effort on investigating and defining the inputs to the network? That is, why investigate the relationship between engine operational parameters and the specific fuel consumption performance? This question was particularly relevant in this work. This was since the purpose of the neural network training was to teach the neural network the general functional relationships between an engine's operational parameters and its fuel consumption performance. Without this generalisation capability, the neural network would serve little purpose in actual application. Therefore, a domain expertise approach was adopted towards the specification of the inputs. These inputs are the several geometric and thermodynamic parameters that in general characterise a diesel engine's specific fuel consumption performance.

The primary geometric property of a diesel engine was found to be its cylinder size, which is defined by its bore and stroke. Of the six selected input parameters to the neural network, five are to a greater and lesser extent a function of the cylinder bore, and four are a function of stroke.

The method of this study to characterise diesel engine specific fuel consumption performance was distinctively different in approach to the other engine performance modelling approaches such as Golverk (1994a/b), and Celik and Arcaklioglu (2004). Golverk focused on normalising the performance maps to the rated mean piston speed, rated mean effective pressure and minimum specific fuel consumption of the engine to define a universal performance map. Celik and Arcaklioglu used engine speed, engine power and cooling water temperature to define a performance map for a specific engine. In this study, not only were engine performance normalising parameters used, but the performance was also characterised by geometric parameters. That is, the effects of cylinder size on fuel consumption performance were included in the input parameter definitions. A further distinctive feature of this work was that domain expertise was applied to qualify the inclusion of the defined input variables to the neural network.

The primary result from this phase of the work was a defined set of parameters that characterise a diesel engine's specific fuel consumption performance. These can form the starting point for further investigations into diesel engine performance prediction models.

Once the parameters defining engine performance were deemed to be adequately defined, the next phase of the research dealt with the correct selection and partitioning of the training and test data.

8.3 Data Selection and Partitioning

Data are the lifeblood of any neural network development. This is because neural networks are trained non-parametrically - they learn from data. Correspondingly, the training set size (or number of training exemplars) directly influences acceptable performance of a neural network. In the context of this research, acceptable performance was considered to be if the neural network could be validated on an engine for which the data was not part of the neural network training. Therefore, selection and correct partitioning of the data into training and test data sets was an important aspect of this work.

In terms of data selection, two requirements were applied to ensure that the neural network was able to capture the problem at hand, that is, be able to generalise on previously “unseen” engine performance data and therefore be validated for the application. These were:

- Investigate methodologies to ensure that there was sufficient training data for the problem at hand.

- Apply a double cross-validation data set or out-of-training test data set on which the models are to be validated.

One may question why there was a need to investigate methodologies for determining the minimum training data set size. There exists fairly simple heuristics that can guide the investigator in terms of determining the minimum number of training data. One of these is given as: – the required number of training examples is directly proportional to the number of weights in the neural network and inversely proportional to the specified or acceptable error. From this, a general guideline is that the number of training exemplars should be ten times the number of free parameters in the network to achieve ninety percent accuracy. The problem was that this type of heuristic was not very helpful in practice. This was because the problem of determining the minimum size of the training data was more complex than the simple application of the heuristic. This complexity arose out of three constraints. In the first instance, the number of weights in the neural network was not known and their number was model dependent. In the second instance, there was the question of what could be deemed to be an acceptable error for the application, and did the training data contain their own error? In the third instance, the number of training exemplars was limited due to data acquisition constraints. These three constraints are typical in neural network development and it is therefore worthwhile investigating methodologies for their resolution. Thus, the data selection and partitioning methodology defined in this work directly defined processes to optimise the neural network development within these constraints.

The second requirement for the data selection and partitioning centred on the issue of whether the training data was in fact relevant to the problem and whether the trained neural network had actually captured the problem. The methodology applied in this work was to use domain expertise for input variable selection as well as a double cross-validation data set (or out of sample test data set). The specific

methodology encompassing double cross-validation was stringent in that it was designed to highlight any deficiency in the training data or define the neural network's generalisation capability after learning from this data. This approach is not unique, however it is distinctive when compared to the approach to data partitioning as used in Golverk, and Celik and Arcaklioglu. In both instances, there was no true out of sample or double-cross validation data set applied in their work.

8.4 Neural Network Selection, Training and Optimisation

The functional type (category), architecture, and typical uses of neural networks are vast. However, due to the nature of the engine performance mapping problem, the category was reduced to the regression or the function approximation type of neural network. Further, due to their relatively simple implementation, the most popular architecture was chosen for the study. That is, feedforward backpropagation in its standard form (non-modular) and in its modular form.

Modular networks have some advantages over non-modular networks such as lower complexity and interference avoidance such as spatial and temporal crosstalk. Due to these factors, modular networks tend to capture problems better (learn better) and have improved generalization capability. This characteristic tends to make them more useful in applications and they have provided solutions to many problems. Therefore, a modular feedforward backpropagation network was developed together with a non-modular feedforward backpropagation network.

Regarding the number of layers for the neural network, the study relied on the proven performance of two hidden layer networks. The first hidden layer tends to extract local features and the second hidden layer tends to extract the global features of the problem being modelled. Therefore, the selection of the neural network type is not a distinctive feature of this work. However, the training methodology applied to the problem was a distinctive aspect of this work.

Training of neural networks is typically resolved using a methodology that encompasses experimental optimisation (trial and error). Experimental optimisation of neural networks has traditionally resulted in neural network development requiring "black-art" skills or highly expert developers to achieve successful results. An alternative to this traditional methodology was proposed and executed in this work to

better formalise the approach to neural network training; specifically when faced with the three practical constraints highlighted in section 8.3.

The methodology proposed for neural network training had the following features:

- The use of a genetic algorithm to globally search for the solution of the optimum number of weights and the learning rate parameters;
- The use of standard error correction learning (backpropagation), together with a local learning algorithm (Delta-Bar-Delta) to establish the network weight values;
- The use of multiple training runs within the genetic algorithm to circumvent the problems associated with network weight initialisation.

The methodology combined probabilistic as well as deterministic rules to learning. Therefore many of the pathologies associated with neural network learning were addressed.

Such a training methodology, however, produces its own constraints and complexity. The main constraint is the processing time associated with the genetic algorithm which is further burdened with multiple training runs. This is highlighted by the following:- for a population of 50 individual genotypes (decoded into phenotypes), 20 multiple runs and 100 epochs per run, one needs 100,000 training epochs per generation. Say, if there are 50 generations of genotypes used to find the solution, one would need $5x10^6$ epochs of training. Therefore, the methodology is practically limited to smaller networks, fast computer processors and efficient neural network development shells. The genetic algorithm also introduces a further level of complexity to the learning process.

Firstly, the backpropagation learning process can affect the efficiency of the genetic algorithm due to inferior learning associated with poor weight initialisation. That is, a perfectly good phenotype could be rejected on unacceptable fitness due to poor weight initialisation. This issue is only partly solved by multiple weight initialisations, since the number of initialisations needs to be constrained due to processing time limitations. Secondly, the network developer is faced with a number of decisions. One needs to define the genetic operations criteria such as selection, reproduction, mutation and stopping. Further, the evolutionary process needs to be constrained by

setting the number of generations, population size, and valid genotype to phenotype decoding.

The network training methodology applied also had a deficiency that can be improved upon. This deficiency lay in the selection of best phenotype performance from multiple training runs. In this study, the mean of each multiple run was compared without consideration of the variance. ANOVA (Analysis of Variance) should have been implemented together with a Tukey-Kramer procedure for comparison and selection of the best mean from the multiple means that were derived in each generation.

It may be argued that the methodology proposed introduces a layer of complexity that is not beneficial to the network development process. However, despite its limitations, the training methodology proved highly successful and is proposed as a superior methodology for neural network training; specifically when faced with limited training data together with the need to develop a neural network that has superior generalisation capability. Further, the proposed training methodology should be used for training modular networks. This is because modular networks are typically much smaller (have less free parameters) than their non-modular counterparts in order to capture the same problem. This was demonstrated in this work with the modular network having 156 weights versus the non-modular network with 449 weights.

The focus of this study was the development of neural network models for diesel engine performance modelling. However, the study was more than simply this. A definitive aspect of the study was the definition of the methodology for verification and validation of the neural network models for their intended purpose.

8.5 Model Verification and Validation

Engineering model validation has changed due to our increased ability to model complex systems, inexpensive computational simulation, and the increased cost of performing real system experiments. As a result, computer simulation is replacing experimental work in many areas. Therefore, verification and validation should always form a significant portion of any study that develops predictive models. Without thorough verification and validation there can be no grounds on which to place confidence in the study's results and correspondingly in the final application of

the model. However, model verification and validation is not straightforward and it was found that it is not often performed as thoroughly as it should be. This is because, for a given problem, there are no formal methods which dictate how a neural network's quality (performance) is to be verified or validated when applied to the problem. There are also several forms (or philosophies) of model validation for a specified application. Therefore the definition of a detailed methodology for the verification and validation of the neural network models for engine performance prediction was a distinctive result of this study. In this study, the verification process ensured that the conceptual neural network models were transformed into computer code with sufficient accuracy, and the validation process tested whether the models were sufficiently accurate for their intended purpose. This ensured that an appropriate model was developed for the intended purpose and goals of the study.

The key concept of validation is the idea of sufficient accuracy. This is since a model is a simplification of the real system, and measuring the real system in itself has its own complexity (for example, measurement or data error). There is no benefit in seeking an exact (100% accurate) model of a system. Rather the validation methodology must allow for the model with sufficient accuracy (how close must close be for the model to be trusted for use?). This was central to the methodology of this study. Defining the sufficient accuracy goals for the models required several performance measures and acceptance criteria. These were as follows:

The mean square error of the model output to the target output is a typical measure of neural network performance. However, it was found that there are practical difficulties in establishing acceptance criteria for the mean square error. Therefore a normalised version of the mean square error was implemented. This normalised mean square error used the nearer specification limit concept that was modified to encompass the definition of an acceptable percentage error level. Here, the acceptable error was equated to the typical level of propagated error that one would expect from the instrumentation measuring the engine performance. This was consistent with the idea that it is reasonable not to expect a higher standard of inference using the model than one could expect from direct measurement of the engine performance. Further, the modified version of the normalised mean square error implemented in this work proposed a good method for setting the error goals for neural network verification and validation. Similar methods to those presented in this study are not readily found in published neural network literature.

With respect to the statistical significance testing (hypothesis testing) performed in this study, there has been much published criticism around the use of hypothesis testing when used on experimental data or observational data. The use of the null hypothesis that states that the model and the real system are the same is clearly false on *a priori* grounds. We know that the model and the real system are not the same and no one really believes the null hypothesis. The problem lies in the use of the level of significance that arbitrarily divides the rejection/non-rejection decision into *significant* and *non-significant* regions. The use of the null hypothesis thus provides no meaningful insight into a problem and has limited utility in the validation process. Therefore, the validation methodology focused on the more useful approach of determining whether or not the differences between the real system and model were *large* enough to affect the model's usefulness in its intended application. This required the use of acceptance or rejection criteria based on the alternative hypothesis that the model and the real system were sufficiently similar. These acceptance criteria were formulated within the black-box approach to validation.

Statistical hypothesis testing (for rejection or non-rejection of the null hypothesis) was therefore performed on the training and cross-validation data only. In this study this was termed model verification. The approach here was to achieve the best possible model generalisation (or generalisation performance) prior to validating the final model on the test data set (the out of training sample or double cross-validation data set). This is because the test data set could not be used to decide between alternative models or training methods and criteria. If the test data set was used for decision making on the non-final models, it would in fact become part of the training data set and there would therefore be no test data set. This is a strict approach to black-box model validation which is not often found in the neural network literature.

A further form of verification, called white-box validation was formulated and applied in this study using the training data set. This took the form of sensitivity analysis. The sensitivity analysis was conducted as both input-response mapping and input-response sensitivity. The primary purpose of the sensitivity analysis performed was to verify that the model response behaved as one would expect to each input parameter. This was verification of the model construct and ensured that the model was correctly formulated and trained. The focus on verification centred on the principle that there was no purpose in attempting to validate a model for an application when it has not been verified as behaving, in general, as expected by a domain expert.

In summary, there were in total 23 individual verification and validation tests proposed for the verification and validation of the models. These included tests for model means within the nearer specification limit and within the confidence level limits. Both of these were formulated to include the specification of an acceptable error and an estimated propagated error. Further, a number of goodness-of-fit tests were formulated as alternatives to the standard coefficient of correlation and the coefficient of determination. These goodness-of-fit tests, namely, the Coefficient of Efficiency and the Index of Agreement in their standard and modified forms were proposed as providing improved analysis of the experimental results. The use of these alternative goodness-of-fit tests and the use of the nearer specification limit together with an estimated propagated and acceptable error is not commonly found in the neural network literature. The verification and validation methodology applied in this work is therefore a unique aspect of this study.

8.5.1 Experimental Results – Model Performance Findings

8.5.1.1 Findings: Verification of Neural Network Learning

The neural network models were found to have met the accuracy goals of the study when tested on the training and cross-validation data sets. However, there were a few findings that are worth further discussion.

There was no significant difference between the mean square error performance (training and cross-validation data) of the neural networks; that is, both networks tended to learn the training data with a similar scale in error. This is despite the expectation that the modular network should have had a superior ability to capture the problem. Further, for both neural networks and using the mean square error performance criteria, there was a significant difference in error when comparing the error on the training data with that on the cross-validation data. This indicated that the generalisation performance of both networks may have been impaired. In this respect it is generally accepted that one can expect a decrease in error performance on the training and cross-validation data sets. However, a drop in performance in the order of 10% to 15% typically indicates deficient learning. It was found, for both trained neural networks, the decrease in error performance was in the order of 22%, thereby indicating deficient learning.

There was a significant difference in the Akaike's Information Criteria (AIC) error performances between the networks, with the non-modular network having had a lower AIC mean than that of the modular network. This indicated that the modular network had a higher complexity (network size) to training set size than that of the non-modular network. This was unexpected in that lower complexity is the reason why modular networks are employed. However, since this difference has a confidence interval in the range of 1.4% to 17.6%, the modular network complexity to training set size is only marginally worse than that of the non-modular network.

The conclusion from the training results is that the non-modular network demonstrated superior training performance over the modular network. Comparing the neural network training performances in fact verifies each model and is a partial form of validation. This is because they have closely matching performances on the training and cross-validation data sets. Hence there is a fair degree of confidence in the neural network validity prior to entering the validation phase.

8.5.1.2 Findings: Input-Response Sensitivity Verification

The sensitivity and input-response mapping of the modular and non-modular networks were found to generally behave as expected - specifically to the mean effective pressure and mean piston speed input parameters. Since these parameters are the primary drivers of engine consumption and define the general shape of the specific fuel consumption map, both trained neural networks appeared to have captured the problem, and were therefore verified. An exception to the expected input-response mapping was found on the non-modular network with the specific power input parameter. For this parameter, the expectation was for an increase in specific fuel consumption for an increase in specific power. However, this parameter had the lowest ranked sensitivity for this model, and therefore retained in the variable set. A partial exception to the input-response mapping was also found on the modular network for the piston area to volume displaced input parameter. The network displayed unexpected behaviour at the lower ratios of piston area to volume displaced.

Diesel engine specific fuel consumption is predominantly a function of mean effective pressure or torque. This can be seen in the polynomial coefficients of the quadratic

model - equation (2.29) and Table 2.3, where the coefficients related to mean effective pressure are 47.3% and 4.5% larger than those related to mean piston speed for the first and second order terms respectively. This can also be seen from the engine performance map surface where, moving along the mean effective pressure axis the gradient is steeper than moving along the engine speed axis. That is, one would expect a significantly larger sensitivity in model response to mean effective pressure than mean piston speed. The modular network displayed a higher sensitivity to mean effective pressure than to mean piston speed with a difference of 229%. This was not found on the non-modular network with a difference of only 7%. The modular network's sensitivity to mean effective pressure was 58% greater than that for the non-modular network. Also the modular network's sensitivity to mean piston speed was 49% lower than that for the non-modular network. The implication is that the shape of the estimated performance map using the neural networks will be different for the same values on these primary input parameters.

Therefore the neural networks, although capturing the problem in general, are not capturing or learning the problem in the same manner. From the sensitivity analysis it is evident that the modular network is capturing the problem better than the non-modular network. This effect may be attributed to the general capability of modular networks and their reduced complexity.

In conclusion, both neural networks were verified by the tests performed in the training phase of the neural network development.

8.5.1.3 Findings: Model Validation for the Application

There were a total of twenty three verification and validation tests performed on the neural network models (this number excludes the four significance tests done during the neural network training). The quadratic model was assumed to be verified by its author, therefore only validation tests were performed on this model. This resulted in a total of eleven validation tests being performed on the quadratic model. The primary objective for validating the quadratic model was to check whether the neural network approach to the problem yielded improved validation performance over the quadratic method.

For the non-modular network, the verification and validation of the alternative hypothesis was accepted for fifteen tests out of twenty three tests (or a score of 65%). In the same manner, that for the modular network was accepted for twenty two tests out of the twenty three tests (or a score of 96%). For the G-model, the alternative hypothesis was not accepted for all eleven tests performed (or a validation score of zero). The quadratic model was found to be invalid for the application, since it had very poor performance in terms of the verification and validation tests. Therefore, both neural network models outperformed the quadratic method in terms of model verification and validation for the application.

Regarding the validation tests failed by neural networks, the non-modular network was not validated on the overall mean within the nearer specification limit or the contour mean. Further, the network failed validation on the interpolated surface maps and specifically the horizontal and vertical interpolated test data. This finding means that the non-modular network could not be used in applications where constant speed and variable torque or constant torque and variable speed are the mode of engine operation.

In terms of the alternative hypothesis acceptance score, the M-model (modular network) can be taken as better validated for the applicant over the F-model (non-modular network). However, when comparing the validation performance of the models, one should not use the test data set to decide on the better validated model. If the test data set is used for alternative model validation performance comparison, it then by definition becomes part of the cross-validation data set. This then defeats the purpose of having an independent test data set. However, since no model parameter tuning (or new learning) was done during the tests and the verification and validation methodologies employed were deemed to be sufficiently critical, here the test data set has been used to select the best model for the application. The best model for the application was found to be the modular neural network.

8.6 Research Application and Benefits

The research completed in this study was focused on providing selection, training and testing methodologies for the development of a neural network to be used in energy monitoring and targeting of diesel engine performance. It has been demonstrated that such a model is key to providing information on the target fuel

consumption of a specified diesel engine. Without such a model, there is simply no means of effectively operating and managing an energy monitoring and targeting process. The result of this is limited information available for continuous improvement of energy efficiency. Therefore, in an industrialised world where there is a dire need to conserve fossil fuels and reduce pollutant emissions, the results of this study are especially important.

The results of this work have already found application on heavy mining vehicles for fuel consumption targeting in a formalised monitoring and targeting process. The process as shown in Figure 7.3 is proving highly beneficial for all the continuous improvement initiatives on the mine. The information provided by the system is assisting the engine manufacturer (Cummins), heavy equipment manufacturer (Komatsu – 730E), fuel supplier (British Petroleum), lubrication supplier (Castrol), and auxiliary systems suppliers (ECS) deliver improved services and performance to the mine. Many decisions made for technological improvements are now based on factual information with the associated financial risks better understood.

8.7 Limitations

The modular neural network validated in this work is limited to normally-aspirated direct injection diesel engines. Also, since neural network based models should not be used for extrapolation, the model should be limited for use on engines for which the geometric and performance parameters fall within the limits defined in Table A.1.1.

Although a double cross-validation test data set was strictly applied for model validation, the test data originated from a manufacturer and engine class that was included in the training data set. Also, due to scarcity of engine map data, only one engine was used to validate the neural network model. The quality of the result (improved trust in the resulting model) would have been greatly increased.

8.8 Further Work

The following are proposed as extensions to this work:

- A broader range of engine physical size, performance and manufacturer should be introduced and the neural network validation confirmed. There should also be a range of engines on which the neural network model should be validated.

- The primary deficiency of this study is that turbocharged engines were not included in the training of the neural network model. This constrains the use of the model in energy monitoring and targeting applications. However, since a definitive methodology for neural network construction, training and validation has been proposed, inclusion of turbocharged or turbocharged and intercooler engines should be readily achievable.

- Multiple training runs or multiple weight initialisations were used for each phenotype of a generation. The resulting fitness evaluation in the genetic algorithm simply used the mean performance of the multiple run. It would be beneficial to test whether the application of ANOVA together with the Tukey-Kramer procedure for multiple comparisons of the means results in an improvement in the algorithm proposed in this study.

- The modular network and the training algorithm could be converted to allow for on-line training. This will greatly improve its scope of application. The conversion can take the form of a DLL (dynamic link library) that is inserted within as an ASP (active server page) or a database (for example, Microsoft Access). Users of the neural network can then provide for semi-automatic on-line training of the network.

- A core objective of energy efficiency improvement is the reduction of pollutant emissions. Developing the neural network model so that emissions can be targeted together with the fuel consumption would be highly beneficial.

Bibliography

Abelson, R.P. 1997. A retrospective on the significance test ban of 1999: If there were no significance tests, they would be invented. In Harlow, L. A., Mulaik, S. A., and Steiger, J. H. (eds.) *What if there were no significance tests?*: Mahwah, NJ: Erlbaum. pp. 117-144.

Ahmed, S.E. 1995. A pooling methodology for coefficient of variation. *The Indian Journal of Statistics, Vol. 57(B):* 57-75.

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle. In Petrov, B.N., and Csaki, F. (eds.) *2nd International Symposium on Information Theory.* Budapest: Akademiai Kiado.

Anderson, D., and McNeil, G. 1992. Artificial neural networks technology: DACS State-of-the-Art Report. ELIN:A011. New York.

Auer, P., Herbster, M., and Warmuth, M.K. 1996. Exponentially many local minima for single neurons. In Touretzky, D.S., Mozer, M., and Hasselmo, M. (eds.) *Advances in Neural Information Systems*. MA: MIT Press.

Balakrishnan, K., and Honavar, V. 1995. Evolutionary design of neural networks: a preliminary taxonomy and guide to literature. Technical Report CS TR 95-01, Department of Computer Science, Iowa State University, Ames, Iowa.

Ballaney, P.L. 1998. *Internal combustion engines: theory and practice*. Delhi: Khanna Publishers.

Barker, T.D. 1982. Engine mapping techniques. *International Journal of Vehicle Design, Vol. 3(2):* 142-152.

Barber, C. B., Dobkin, D.P., and Huhdanpaa, H.T. 1996. The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software, Vol. 22(4):* 469-483.

Battaglia, G.J. 1996. Mean square error. *AMP Journal of Technology, Vol. 5,* 31-36.

Baum, E.B., and Haussler, D. 1989. What size net gives valid generalization? *Neural Computation, Vol. 1:* 151-160.

Bishop, C.M. 1994. Mixture density networks: Technical Report NCRG/94/001. *Neural Computing Research Group*. Birmingham: Aston University.

Bishop, C.M. 1996. Theoretical foundations of neural networks: Technical Report NCRG/96/024. *Neural Computing Research Group*. Birmingham: Aston University.

Bradley, M.L., Raghavan, R., and Slawny, J. 1989. Backpropagation fails to separate where perceptrons succeed. *IEEE Trans. Circuits Systems, Vol. 36:* 665-674.

Boers, E.J.W., and Kuiper, H. 1992. *Biological metaphors and the design of modular artificial neural networks*, MSc thesis, Leiden University, Netherlands.

BS 5514: Part 3: 1990 / ISO 3046-3: 1989. Reciprocating internal combustion engines: *Performance: Specification for test measurements*. London: British Standards Institute.

BS 5703-1:2003. Guide to data analysis and quality control using cusum techniques. *Uses and value of cusum charts in business, industry, commerce and public service*. London: British Standards Institute

BS 5703-2:2003. Guide to data analysis and quality control using cusum techniques. *Introduction to decision-making using cusum technique.* London: British Standards Institute

BS 5703-3:2003 Guide to data analysis and quality control using cusum techniques. *Cusum methods for process/quality control using measured data.* London: British Standards Institute

Carson, J.S. 1986. Convincing users of model's validity is challenging aspect of modeler's job. *Industrial Engineering, Vol.18(6):* 74-85.

Celik, V., and Arcaklioglu, E. 2004. Performance maps of a diesel engine. *Applied Energy*. Elsevier Ltd. Available: Science Direct, Article No 0306-2619 [Accessed 10 August 2004].

Chester, D.L. 1990. Why two hidden layers are better than one. *IJCNN-90-WASH-DC, Vol. 1*. Washington: Lawrence Erlbaum, p. 265-268.

Cohen, P.R. 1995. *Empirical methods for artificial intelligence*. Cambridge, MA: MIT Press.

Coulibaly, P., Anctil, F., and Bobée, B. 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology, Vol.230:* 244-257.

Cybenko, G. 1989. Approximations by superpositions of a sigmoidal function. *Mathematics, Control, Signals & Systems, Vol. 2:* 304-314.

Davis, P.K. 1992. Generalizing concepts of verification, validation and accreditation (VV&A) for military simulation. R-4249-ACQ, RAND, Santa Monica, CA.

De Garis, H. 1990. Genetic programming: modular neural evolution for Darwin machines. In: *Proceedings of the International Joint Conference on Neural Networks, Washington DC.* NJ: Lawrence Erlbaum, p. 194-197.

Devin, D.W.1982. *Energy – its physical impact on the environment*. USA: Wiley.

Dodd, N. 1990. Optimization of network structure using genetic algorithms. In Widrow, B., and Angeniol, B. (eds.). *Proceedings of the International Neural Network Conference, INNC-90-Paris.* Dordrecht: Kluwer, p. 693–696.

Efron, B., and Tibshirani, R. 1993. *An introduction to bootstrap*. New York: Chapman & Hall.

Egmont-Petersen, M., Talmon, J.L., Brender, J., and McNair, P. 1994. On the quality of neural net classifiers. *Artificial Intelligence in Medicine, Vol. 6*: 359-381.

Ely, C.G. 1998. Engine optimization program lowers maintenance costs. *Pipeline and Gas Journal, Vol. 225(10):* 40.

Feelders, A., and Verkooijen, W. 1995. Which method learns most from data*?, Proceedings of the fifth international workshop on AI and Statistics*, January 1995. Fort Lauderdale, Florida, p.219-225.

Ferguson, C.R., and Kirkpatrick, A.T. 2001. *Engines: Applied Thermoscience.* 2nd Edition. New York: John Wiley & Sons Inc

Flexer, A. 1995. Statistical evaluation of neural network experiments: minimum requirements and current practice, *Technical Report OEFAI-TR-95-16.* Vienna: The Austrian Research Institute for Artificial Intelligence.

Frey, H.C., and Patil, S. 2001. Identification and review of sensitivity analysis methods. *Proceedings of NCSU/USDA Workshop on Sensitivity Analysis Method*.

Funahashi, K. 1989. On the approximate realization of continuous mappings by neural networks. *Neural Networks, Vol. 2:* 183-192.

Glorfeld, L.W. 1996. A methodology for simplification and interpretation of backpropagation-based neural network models. *Expert Systems with Applications, Vol.10(1):* 37-54.

Goldberg, D.A. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison Wesley.

Golverk, A.A. 1994a. The method for development of a diesel engine universal performance map. *SAE Technical Paper 941928.* Warrendale, Pennsylvania, USA: Society of Automotive Engineers.

Golverk, A.A. 1994b. Diesel engines fuel economy characteristics. *SAE Technical Paper 941730.* Warrendale, Pennsylvania, USA: Society of Automotive Engineers.

Golverk, A.A. 1994c. Tractor diesel engine performance under variable loading. *SAE Technical Paper 940209.* Warrendale, Pennsylvania, USA: Society of Automotive Engineers.

Grove, D.M., and Davis, T.P. 1995. *Engineering, Quality and Experimental Design.* U.K.: Longman Group.

Guyon, I. 1991. Applications of neural networks to character recognition. *International Journal of Pattern Recognition and Artificial Intelligence, Vol. 5:* 353-382.

Hagan, M.T., Demuth, H.B., and Beale, M.H. 1996. *Neural Network Design*. Boston, MA: PWS Publishing.

Happel, B.L.M., and Murre, J.M.J. 1994. The design and evolution of modular neural network architectures. *Neural Networks, 7:* 985-1004.

Harp, S.A., Samad, T., and Guha, A. 1989. Towards the genetic synthesis of ANN. In *proceedings of the 3rd International Conference on Genetic Algorithms (ICGA'89)*, p. 360-369.

Harris, H.D., and Pearce, F. 1990. A universal mathematical model of diesel engine performance. *Journal of Agricultural Engineering Research, Vol. 47:* 165-176.

Haykin, S. 1994. *Neural Networks: a comprehensive foundation*. USA: Macmillan.

Heisler, H. 1995. *Advanced engine technology*. London: Edward Arnold.

Heywood, J.B. 1988. *Internal combustion engine fundamentals.* New York: McGraw-Hill.

Hills, R.G., and Trucano, T.G. 1999. *Statistical Validation of Engineering and Scientific Models: Background*. SAND99-1256. Las Cruces, New Mexico: Department of Mechanical Engineering, New Mexico State University.

Holland, J.H. 1976. *Adaptation in natural and artificial systems*. Ann Arbor, MI: The University of Michigan Press.

Holliday, T., and Lawrence, A.J. 1998. Engine-mapping experiments: A two-stage regression approach. *Technometrics, Vol. 40(2):* 120-127.

Hornik, K., Stinchcombe, M., and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks, Vol. 2:* 359-366.

Hush, D.R., and Horne, B.G. 1993. Progress in supervised neural networks: what's new since Lippmann? *IEEE Signal Processing Magazine, Vol. 10:* 8-39.

Hussain, T.S. 1995. Modularity within neural networks. Ph.D. thesis, Queens University, Ontario.

Iglewicz, B. 1983. Robust scale estimators and confidence intervals for location. In Hoaglin, D.C., Mosteller, M. and Tukey, J.W. (eds.) *Understanding Robust and Exploratory Data Analysis*. NY: Wiley.

Jordan, M.I. 1995. Why the logistic function? A tutorial discussion on probabilities and neural networks, MIT Computational Cognitive Science Report 9503. MA: MIT.

Jordan, M.I., and Bishop, C.M. 1996. *Neural Networks*. In: Tucker, A. (ed.) *CRC Handbook of Computer Science.* Boca Raton, Florida: CRC Press. pp. 6-7.

Jacobs, R.A. 1988. Increased rates of convergence through learning rate adaptation. *Neural Networks, Vol. 1:* 295-307.

Jacobs, R.A., Jordan, M.I., and Barto, A.G. 1990. Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks. *Cognitive Science, Vol.15:* 219–250.

Kates, E.J., and Luck, W.E. 1975. *Diesel and high compression gas engines*. 3rd Edition. Chicago: American Technical Society.

Kecman, V. 2000. *Learning and soft computing with support vector machines, neural networks, and fuzzy logic models*. USA: MIT Press.

KHD Deutz – Deutz AG, Deutz-Mülheimer Str. 147-149, D-51057 Köln: http://www.deutz.de

Kleijnen, J. P., Bettonvil,B., and Groenendaal, W. (1998). Validation of trace-driven simulator models: A novel regression test. *Management Science, Vol. 44(6).*

Law, A.M., and McComas, M.G. 2001. How to build valid and credible simulation models. In: Peters, B.A., Smith, J.S., Medeiros, D.J., and Rohrer, M.W. (eds.). *Proceedings of the 2001 Winter Simulation Conference,* p. 22-29.

Lawrence, S., Giles, C.L., and Tsoi, A.C. 1997. Lessons in neural-network training: overfitting may be harder than expected. In: *Proceedings of the 14th National Conference on Artificial Intelligence, AAA-97*. Menlo Park: California, AAAI Press, p. 540-545

Le Cun, Y. A. 1988. Theoretical framework for back-propagation. In *Proceedings of the 1988 Connectionist Models Summer School.* San Mateo: Morgan Kaufmann, p. 21-28.

Lee, J., Sheu, B.J., Fang, C., and Chellappa, R. 1993. VLSI neuroprocessors for video motion detection. *IEEE Transactions on Neural Networks, Vol.4:* 178-191.

Legates, D.R. 1999. Evaluating the use of "goodness of fit" measures in hydrological and hydroclimatic model validation. *Water Resources Research, Vol. 35:* 233-241.

Lock, D.F. 1998. *Using genetic algorithms to build, train and optimize neural networks.* MSc Thesis, Department of Computer Science, University of Bristol, Bristol, UK.

Lock, D., and Giraud-Carrier, C. 1998. *Evolutionary programming of near-optimal neural networks.* Department of Computer Science, University of Bristol, Bristol, UK.

Macfarlane, D. 1992. *A practical investigation of parallel genetic algorithms and their application to structuring of artificial neural networks*. Ph.D. thesis, University of Buckingham, Buckingham, UK.

Makartchouk, A. 2002. *Diesel engine engineering: thermodynamics, dynamics, design, and control.* New York: Marcel Dekker.

Matteucci, M. 2002. EleaRNT: Evolutionary learning of rich neural topologies, *Technical Report No. CMU-CALD-02-103*. Centre for automated Learning & Discovery, Carnegie Melon University.

McAulay, K., Wu, T., Chen, S., Borman, G., Myers, P., and Uyehara, O. 1965. Development and evaluation of the simulation of the CI engine. *Society of Automotive Engineers, Paper No. 650451.*

Mendel, J.M., and McLaren, R.W. 1970. Reinforcement-learning control and pattern recognition systems. In Mendel, J.M., and Fu, K.S. (eds.). *Adaptive, Learning, and Pattern Recognition Systems: Theory and Applications*. New York: Academic Press, p. 287-318

Michie, D., Spiegelhalter, D.J., and Taylor, C.C. (ed.) 1994. *Machine Learning, Neural and Statistical Classification*. England: Ellis Horwood.

Miller, G.F., Todd, P.M., and Hegde, S.U. 1989. Designing neural networks using genetic algorithms. In: Schaffer, J.D. (ed.) *Proceeding of the 3rd International Conference on Genetic Algorithms and their Applications.* San Mateo: Morgan Kaufmann, p. 379-384.

Mitchell, P. L. 1997. Misuse of regression for empirical validation models. *Agricultural Systems, Vol. 54(3):* 313-326.

Montgomery, D.C. 2000. *Design and analysis of experiments.* 5th Edition. New York: John Wiley.

Nash, J.E., and Sutcliffe, J.V. 1970. River flow forecasting through conceptual models: a discussion of principles. *Journal of Hydrology, Vol.10:* 282-290.

Orponen, P. 2000. An overview of the computational power of recurrent neural networks. *Finnish AI Conference*. Helsinki.

Orr, G.B., and Mueller, K.R. (eds.). 1998. *Neural networks: tricks of the trade.* Berlin Springer.

Parker, D.B. 1985. Learning logic: casting the cortex of the human brain in silicon, *Technical Report TR-47.* MA, MIT: Centre for Computational Research in Economics & Management Science.

Reckhow, K. H., Clements, J.T., and Dodd, R.C. 1990. Statistical evaluation of mechanistic water quality models. *Journal of Environmental Engineering, Vol. 116(2):* 250-268.

Roberts, P. 2004. *The End of Oil; the decline of the petroleum economy and the rise of a new order.* London: Bloomsbury.

Robinson, S. 1999. Simulation verification, validation and vonfidence: a tutorial. *Transactions of the Society for Computer Simulation International, Vol.16(2):* 63-69.

Robinson, S. 2004. *Simulation: the practice of model development and use*. UK: John Wiley & Son, Inc.

Rueckl, J.G., Cave, K.R. and Kosslyn, S.M. 1989. Why are 'what' and 'where' processed by separate cortical visual systems?: a computational investigation. *Journal of Cognitive Neuroscience, Vol. 1:* 171- 186.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing: explorations in the microstructure of cognition. Vol. 1.* Cambridge: MIT Press. Chapter 8

Russo, A.P. 1991. Neural networks for sonar signal processing, Tutorial No.8. *IEEE Conference on Neural Networks for Ocean Engineering*. Washington, D.C.

Rykiel, E. J. 1996. Testing ecological models: the meaning of validation. *Ecological Modelling, Vol. 90:* 229-244.

SAE Surface Vehicle Standard. 1995. *Procedure for mapping performance – spark ignition and compression ignition engines,* J1312. Warrendale, Pennsylvania, USA: Society of Automotive Engineers.

Saltelli, A., Chan, K., and Scott, E.M. 2000. *Sensitivity analysis*. USA: John Wiley and Sons.

Salustowicz, R. 1995. *A genetic algorithm for the topological optimization of neural networks*, Material No. 127242. Berlin: Technical University of Berlin.

Schiffmann, W., Joost, M., and Werner, R. 1993. *Comparison of optimised backpropagation algorithms.* Presented at ESANN1993. Brussels.

Siegelmann, H.T., and Sontag, E.D. 1999. Turing computability with neural networks. *Applied Mathematics Letters, Vol. 4:* 77-80.

Sima, J., and Orponen, P. 2001. Computing with continuous-time Liapunov systems. *33rd ACM,* Stockholm.

Solla, S.A. 1989. Learning and generalization in layered neural networks: the contiguity problem. In: Personnas, L., and Dreyfus, G. (eds.) *Neural networks: from models to applications: I.D.S.E.T.* Paris, p. 168–177

Sontag, F.D. 1992. Feedback stabilization using two-hidden-layer nets. *IEEE Transactions on Neural Networks, 3:* 981-990.

Soulié, F. 1994. Integrating neural networks for real world applications. In: Zuranda, Marks, and Robinson (eds.). *Computational Intelligence: Imitating Life.* New York: IEEE Press.

Stone, R. 1999. *Introduction to Internal Combustion Engines*. 3rd Edition. London: MacMillan Press Ltd.

Sung, A. 1998. Ranking importance of input parameters of neural networks. *Expert Systems with Applications, Vol. 15:* 405-411.

Tabakov, P. 2001. Multi-dimensional design optimisation of laminated structures using an improved genetic algorithm. *Composite Structures, Vol 54:* 349-354.

Tao, C. 2000. *Sensitivity analysis of neural control.* I-Lan: Taiwan: Dept. of Electrical Engineering, National I-Lan Institute of Technology.

Tarassenko, L. 1998. *A guide to neural computing application: neural computing applications forum.* London: Arnold.

Taylor, C.F. 1985. *The internal-combustion engine in theory and practice, Volume I: thermodynamics, fluid flow, performance*. 2nd Edition. Massachusetts: MIT Press.

Taylor, W.A. 2000. *Change point analysis: a powerful new tool for detecting changes.* Round Lake, USA: Baxter Healthcare Corporation.

Thomas, F.J., Ahluwalia, J.S., Shamah, E., and Van der Horst, G.W. 1984. Medium-speed diesel engines part1: design trends and the use of residual/blended fuels. *Paper 84-DGP-15.* USA: ASME

Thimm, G., and Fiesler, E. 1994. High order and multilayer perceptron initialization. *IDIAP Technical Report, Number 94-07*. Valais: Switzerland.

Thimm, G., and Fiesler, E. 1997. Optimal Setting of Weights, Learning Rate and Gain. *IDIAP Research Report, RR97-04*. Valais: Switzerland.

Valiant, L. 1988. Functionality in neural nets, learning and knowledge acquisition. In *Proceedings of AAAI*. p. 629-634.

van der Smagt, P., and Hirzinger, G. 1998. Solving the ill-conditioning in neural network learning. In: Orr, G.B., and Mueller, K.R. (eds.). *Neural networks: tricks of the trade.* Berlin Springer. p. 193-206.

Von Schnurbein, E., and Bucher, J. 1981. Experience with the rating and operation of medium-speed, four stroke engines under extreme site conditions. *CINIAC paper D64.*

Walczak, S., and Cerpa, N. 1999. Heuristic principles for the design of artificial neural networks. *Information and Software Technology, 41(2):* 109-119.

Wark, K., and Richards, D. 1999. *Thermodynamics.* 6th Edition. USA: McGraw-Hill.

Webster, R., and McBratney, A.B. 1989. On the Akaike Information Criterion for choosing models for variograms of soil properties. *Journal of Soil Science, Vol. 40:* 493-496.

Weisstein, E.W., 2004. Hypothesis testing. *MathWorld* – A Wolfram Web Resource. Available at: http://mathworld.wolfram.com/

Werbos, P.J. 1974. *Beyond regression: new tools for prediction and analysis in the behavioural Sciences.* Ph.D. thesis, Harvard University, Cambridge, M.A.

White, H. 1992. *Artificial neural networks: approximation and learning theory*. USA: Blackwells.

Widrow, B., and Lehr, M.A. 1990. 30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation. *Proceedings of the IEEE, Vol. 78:*1415-1442.

Wieland, A., and Leighton, R. 1987. Geometric Analysis of Neural Network Capabilities. *1st IEEE International Conference on Neural Networks, Vol 3.* San Diego:CA, 385-392.

Willmott, C.J. 1981.. On the validation of models. *Physical Geography, Vol. 2:* 184-194.

Willmott, C.J., Ackleson, S.G., Davis, R.E., Feddema, J.J., Klink, K.M., Legates, D.R., O'Donnell, J.O., and Rowe, C.M. 1985. Statistics for the evaluation and comparison of models. *Journal of Geophysics Res., Vol. 90:* 8995-9005.

Whitley, D. 1989. The genitor algorithm and selection pressure: why rank-based allocation of reproduction trials are best. In: *Proceedings of the Third International Conference on Genetic Algorithms and their Applications*. MIT, p. 116-121.

Whitley, D., and Starkweather, T. 1990. Optimizing small neural networks using a distributed genetic algorithm. In: *Proceedings of International Joint Conference on Neural Networks, Vol. 1*. Hillsdale: Lawrence Erlbaum, p. 206-209

Yao, X. 1992. *A review of evolutionary artificial networks*. Victoria: Commonwealth Scientific and Industrial Research Organization.

Yao, X. 1999. Evolving artificial neural networks. *Proceedings of the IEEE, Vol. 87.*

Yao, X., and Liu, Y. 1997. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks, Vol. 8(3):* 694-713.

Appendix A.1

Engine Parameters

Table A.1.1: Engine Parameters: Engine Parameters used for NN Training and Testing

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Reference	Type	Model Number	Engine Number	Bore (mm)	Stroke (mm)	Cylinders	Rated Power (kW)	Total Cylinder Area (m^2)	Total Swept Volume (dm^3)	Piston Area to Volume (m)	Rated Specific Power (kW/m^2)	Standard Rated Specific Power (kW/m^2)	Map Min SFC (g/kW.h)	Map Rated II SFC (g/kW.h)	Standard Rated SFC (g/kW.h)
214_2986	KHD Deutz	F6L413F	1	125	130	6	130	0.0736	9.572	7.692	1766	1706	211	235	246
414_2376	KHD Deutz	F6L413FW	2	125	130	6	112	0.0736	9.572	7.692	1521	1706	230	276	268
223_0681	KHD Deutz	F4L912	3	100	120	4	54	0.0314	3.770	8.333	1719	1784	215	238	255
223_0677	KHD Deutz	F6L912	4	100	120	6	81	0.0471	5.655	8.333	1719	1784	212	238	258
223_5574	KHD Deutz	F4L913	5	102	125	4	64	0.0327	4.086	8.000	1958	1776	215	247	263
223_5576	KHD Deutz	F6L913	6	102	125	6	96	0.0490	6.128	8.000	1958	1776	215	233	249
MAN_15-21	MAN		7	102	100	8	122	0.0654	6.537	10.000	1866	1776	220	242	253
MAN_15-22	MAN		8	76.5	80	4	36	0.0184	1.471	12.500	1960	1882	246	290	288

Notes:

See Appendix A.2 for the engine performance maps.
All engines are normally aspirated.
Column 13: For normally aspirated engines the Standard Specific Power is given by $4480 \times b^{-0.2}$
Column 14: The standard rated Specific Fuel Consumption is given by $570 \times b^{-0.196} + \left(Q_{c_{rated}} - Q_{c_{min}}\right)$
Engine number 4 is the final test engine.

Appendix A.2

Engine Performance Maps

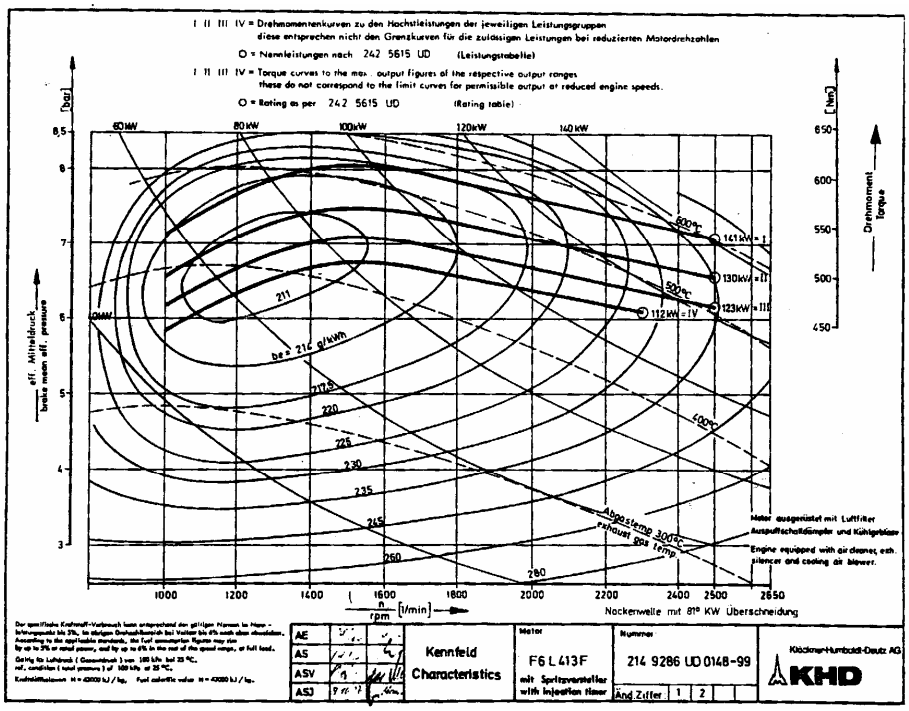


Figure A.2.1: Engine No.1 Performance Map – KHD Deutz

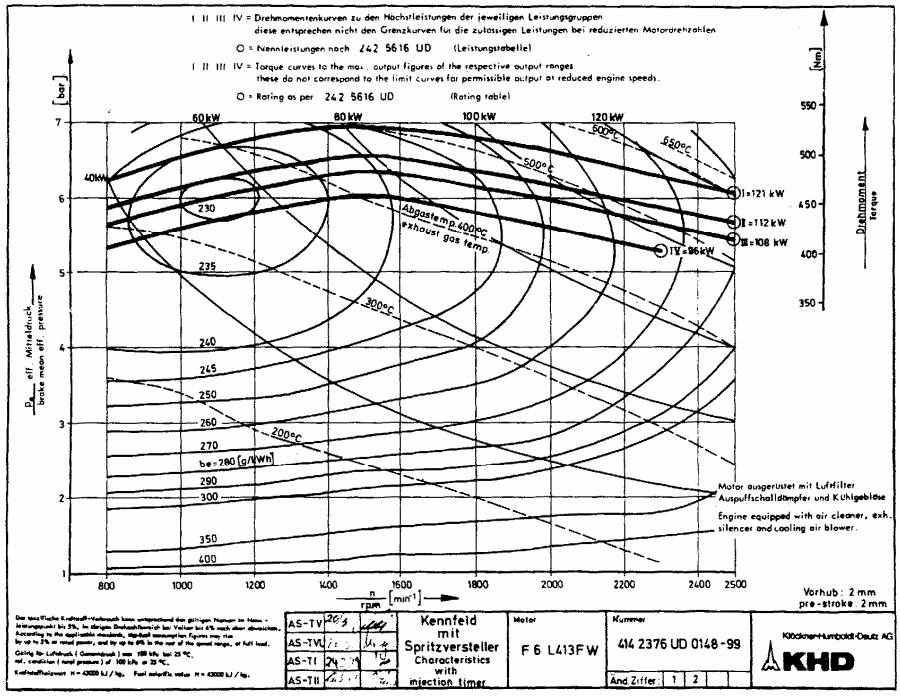


Figure A.2.2: Engine No.2 Performance Map – KHD Deutz

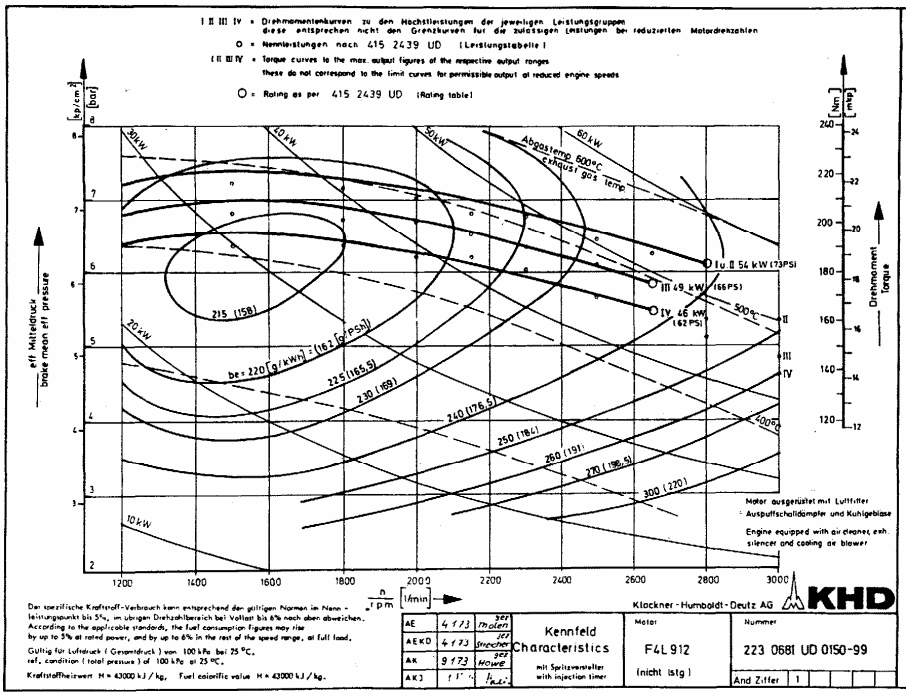


Figure A.2.3: Engine No.3 Performance Map – KHD Deutz

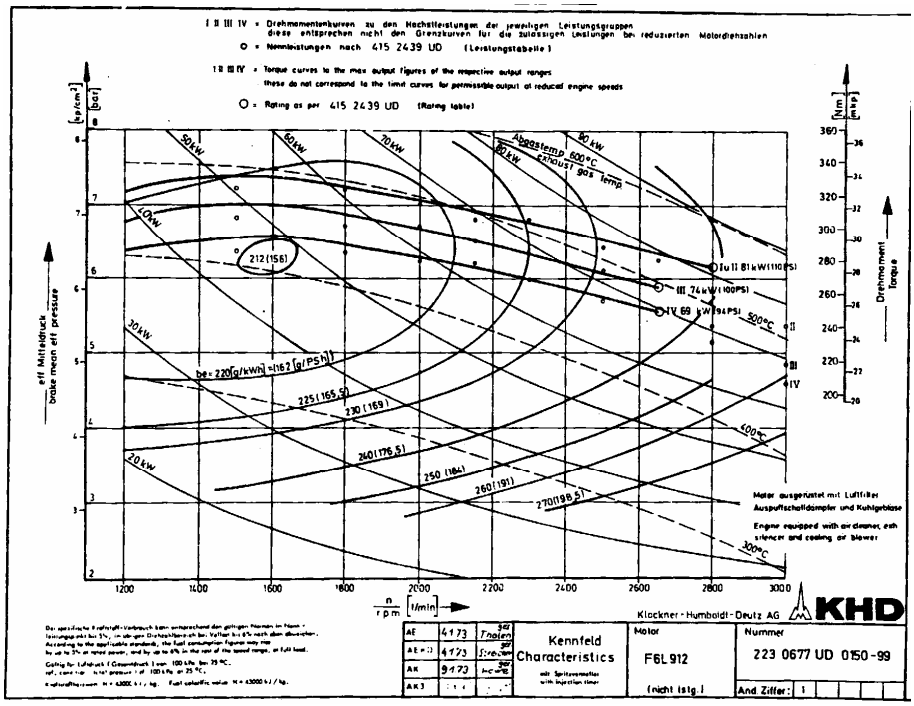


Figure A.2.4: Engine No.4 Performance Map – KHD Deutz
[Model Validation Test Engine]

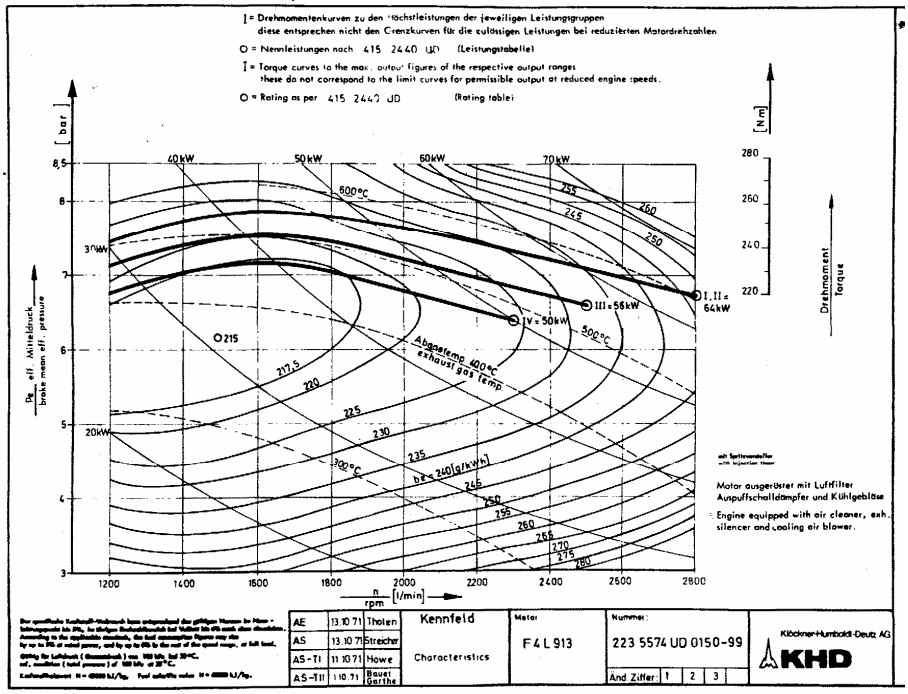


Figure A.2.5: Engine No.5 Performance Map – KHD Deutz

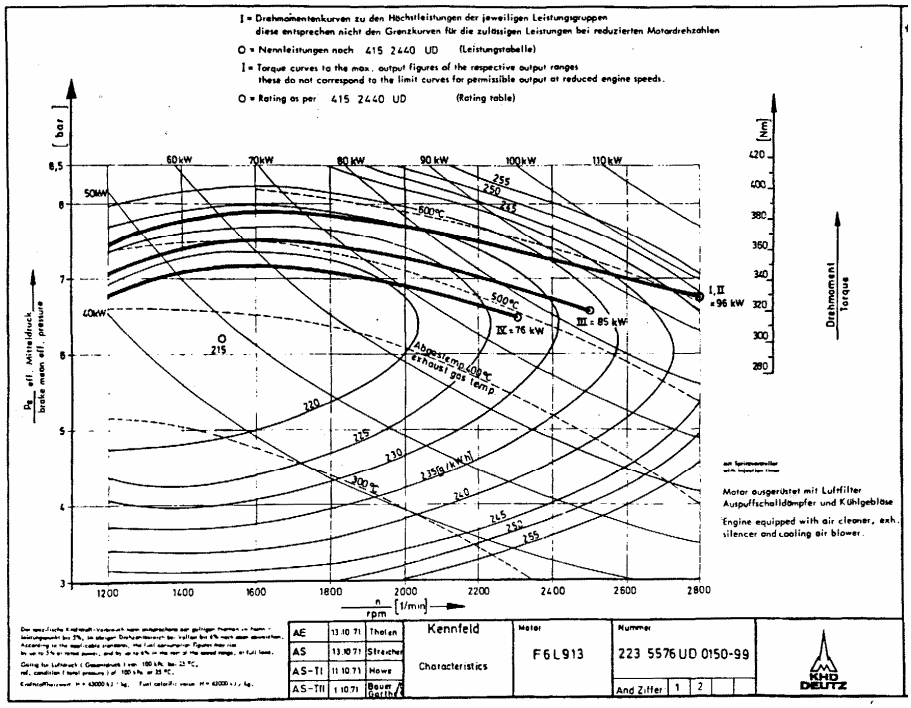


Figure A.2.6: Engine No.6 Performance Map – KHD Deutz

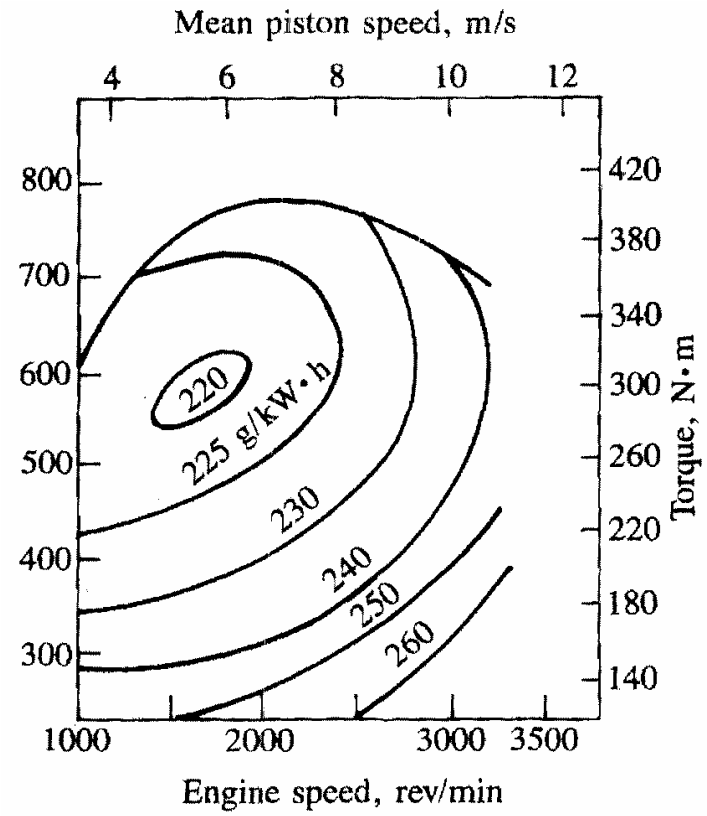


Figure A2.7: Engine No.7 Performance Map – MAN (Heywood,1985)

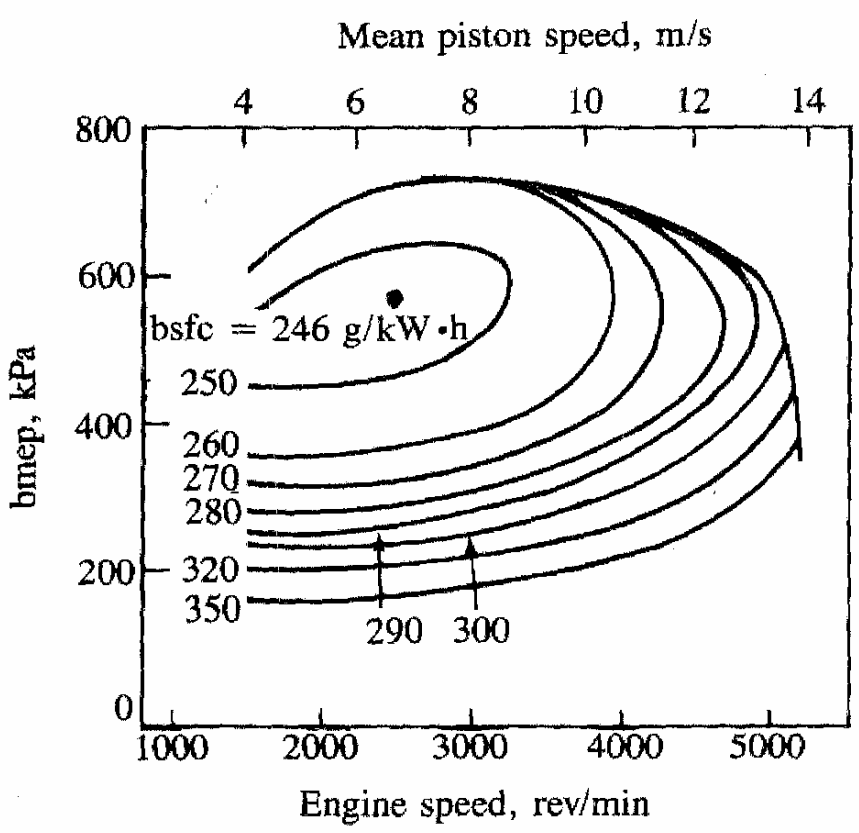
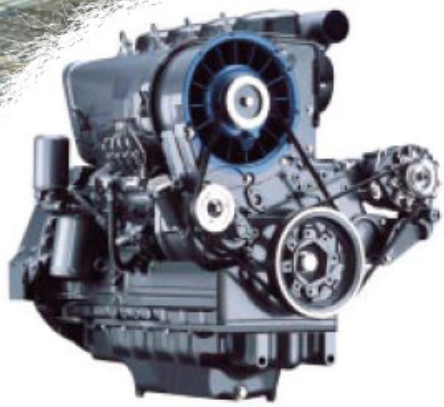


Figure A2.8: Engine No.8 Performance Map – MAN [Heywood, 1985]

912. The engine for construction equipment.

DEUTZ

24 - 82 kW at 1500 - 2500 min^{-1}



These are the characteristics of the 912:

Air-cooled 3-, 4-, 5-, 6-cylinder naturally aspirated in-line-engines.
Direct injection.
Advanced injection and combustion system.
PTOs via gear, V-belt and crankshaft.
Extremely compact design.
High torque at low speeds.
Modular system with single cylinder arrangement and high degree of parts commonality.
Customized component system with many different peripheral parts.
Cold-starting ability even under extreme climatic conditions.

These are the benefits for you:

- Fast response to load changes.
- Low noise emission, high cost savings thanks to less noise insulation requirement.
- Low operating costs thanks to lower fuel consumption and long maintenance intervals with reduced maintenance requirement.
- Excellent smooth-running characteristics thanks to low engine vibrations.
- Minimal environmental impact. Meets exhaust regulation EU-RL 97/68.
- Extremely reliable and durable.
- Easy-to-install unit (engine with integrated cooling system).

Engine description

Cooling system:	Air-cooled with integrated axial-flow blower
Crankcase:	Grey cast iron
Cylinder head:	Aluminium single cylinder heads
Valve arrangement/ timing:	Overhead valves in the cylinder head, one inlet and one exhaust valve per cylinder, actuated from gear-driven camshaft via tappets, push-rods and rocker arms
Piston:	Three-ring piston: two compression rings and one oil scraper ring
Piston cooling:	Oil spray via nozzle
Crankshaft:	Drop-forged steel crankshaft with bolted counterweights
Connecting rod:	Drop-forged steel rod, diagonally split
Main and big end bearings:	Ready-to-install tri-metal plain bearings
Camshaft:	Steel, seated in bi-metal bearing on the blower side
Lubrication system:	Forced-feed circulation lubrication with rotary pump which feeds both lubricating and heating systems (if heating is fitted)
Engine oil cooler:	Integrated aluminium cooler
Oil cooler thermostat:	Oil cooler flow thermostatically controlled on engines with heating system
Lube oil filter:	Paper-type micro-filter as replaceable-cartridge full flow filter
Injection pump/ governor:	In-line injection pump with mechanical centrifugal governor
Injection nozzle:	Five-hole-nozzle
Fuel filter:	Replaceable cartridge
Starter motor:	12V; 2,7 kW (Standard)
Alternator:	Three-phase alternator, 14 V; 55A (Standard)
Heating system:	Optional connection for cab heating
Options:	Intake manifold connections, exhaust manifold connections, compressors, hydraulic pumps, engine mounts rigid and flexible, oil pans, SAE 1/2/3/4 flywheel housings, three-phase alternators 12 and 24 Volt, integrated hydraulic oil cooler, cooling fans controlled by exhaust thermostat

Test Engine

Technical data

Engine type		F3L912	F4L912	F5L912	F6L912
Number of cylinders		3	4	5	6
Bore/stroke	mm	100/120	100/120	100/120	100/120
Displacement	l	2.83	3.77	4.71	5.66
Compression ratio		18	18	18	18
Max. rated speed	min^{-1}	2500	2500	2500	2500
Mean piston speed	m/s	10	10	10	10
Power ratings for construction equipment engines[1]					
Power ratings for automotive engines[2]	kW	40	54	68	82
at speed[3]	min^{-1}	2500	2500	2500	2500
Mean effective pressure	bar	6.79	6.88	6.93	6.96
Power ratings for industrial engines[4]					
highly intermittent operation	kW	40	54	68	82
at speed	min^{-1}	2500	2500	2500	2500
Mean effective pressure	bar	6.79	6.88	6.93	6.96
intermittent operation[4]	kW	38	51	65	78
at speed	min^{-1}	2500	2500	2500	2500
Mean effective pressure	bar	6.45	6.50	6.62	6.62
Max. torque	Nm	185	247	308	370
at speed	min^{-1}	1450	1450	1450	1450
Minimum idle speed	min^{-1}	650	650	650	650
Specific fuel consumption[5]	g/kWh	225	225	225	225
Weight to DIN 70020, Part 7A[6]	kg	270	300	380	410

Model designation

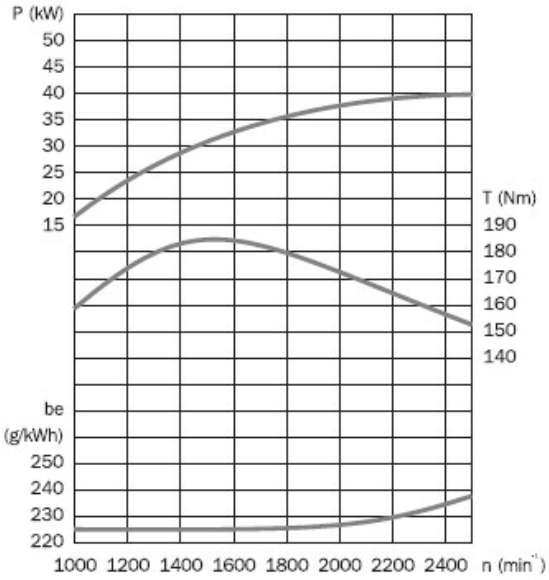
F 6 L 912

- 912 — Engine family designation
- L = Air-cooled
- 6 — Number of cylinders
- F = High-speed four-stroke engine

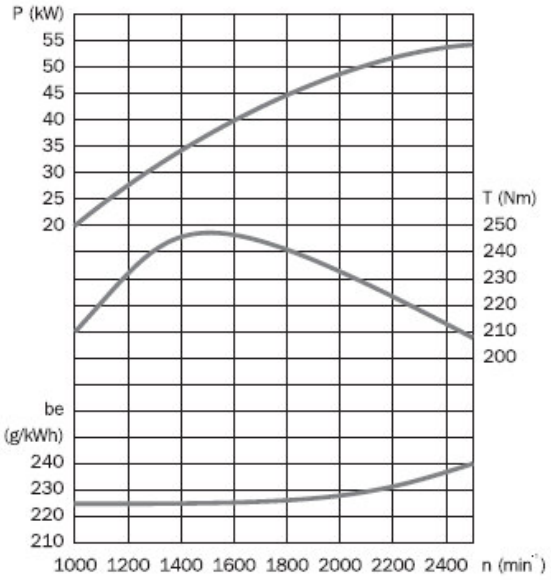
1) Power ratings without deduction fan power requirement, incl. cooling system, meeting exhaust emission limits of EU-RL 97/68.
2) Power ratings to DIN ISO 1585, EG-RL80/1269/EWG ECE-R 24
3) Power ratings for non-mentioned speeds upon request.
4) Power to DIN ISO 3046/1 (IFN). The fuel stop IFN power is an ISO net power at flywheel under reference conditions with all essential auxiliaries driven by the engine.
5) At optimal operating point. Specific fuel consumption based on diesel fuel with a specific gravity of 0,835 kg/dm³ at 15°C.
6) Without starter motor/alternator, radiator and liquids, however with flywheel and flywheel housing and complete integrated cooling system.

The values given in this data sheet are for information purposes only and not binding.
The information given in the offer is decisive.

Standard engines

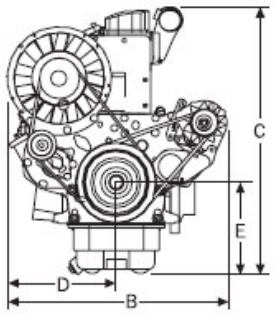


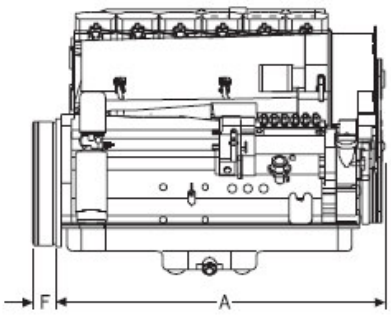
F3L912



F4L912

Dimensions



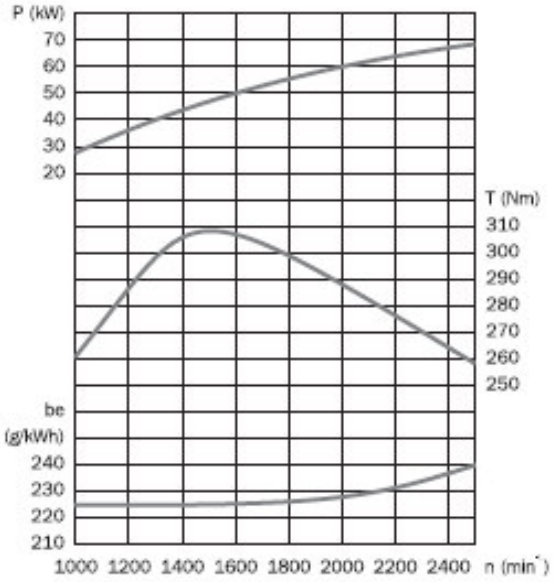


Engine		A	B	C	D	E	F
F3L912	mm	589	679	796	329	268	108
F4L912	mm	719	679	796	329	268	88
F5L912	mm	866	679	833	329	305	88
F6L912	mm	996	679	806	329	278	88

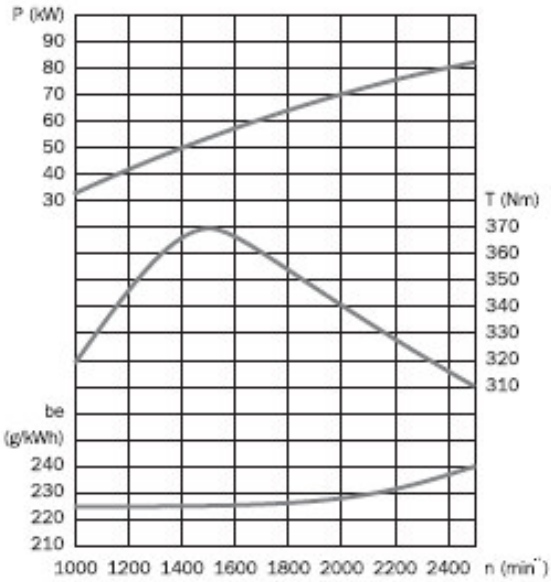
*) with standard flywheel, incl. cooling system
**) with standard oil pan, oil sump central

Test
Engine

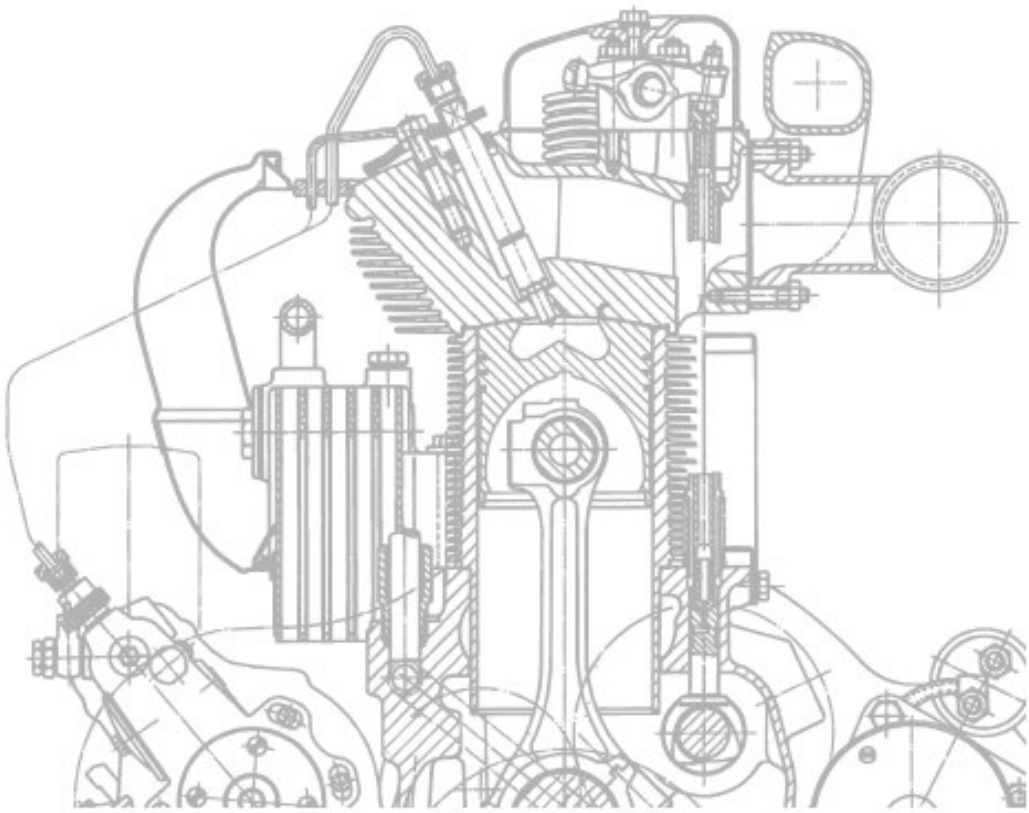
Standard engines



F5L912



F6L912



Appendix A.3

Statistical Methods

(Description and Calculations)

A.3.1 Specific Fuel Consumption Propagated Error

Estimation of the acceptable accuracy for the trained neural network is detailed here. The acceptable error is based on the principle of error propagation of the instrument error. Where the instruments are used to measure the variables needed for input to the neural network models.

From equation(2.18), the specific fuel consumption is given by:

$$Q_c = \frac{\dot{m}_c}{\dot{W}_b} = \frac{\dot{m}_c}{2\pi N \tau} \tag{A.3.1}$$

where $\dot{m}_c$ is the fuel mass flow rate, N is the engine speed and τ is the engine crankshaft torque.

The fuel mass flow $\dot{m}_c$ is rarely measured directly by a mass flow meter and it is common to use a volumetric positive displacement meter that is fuel density compensated. Since fuel density is a function of temperature, the common equation for mass flow is given by:

$$\dot{m}_c = \dot{v}_c(\rho_{\text{ref}} - aT_c) \tag{A.3.2}$$

where $\dot{v}_c$ is the volumetric flow rate as measured by the fuel flow meter, ρ_{ref} is the fuel reference density, a is a temperature compensating coefficient and T_c is the temperature of the fuel at the fuel meter. Therefore:

$$Q_c = \frac{\dot{v}_c(\rho_{\text{ref}} - aT_c)}{2\pi N \tau} = \frac{\dot{v}_c \rho_c}{2\pi N \tau} \tag{A.3.3}$$

It is clear that specific fuel consumption is a function of four measured variables. Therefore $Q_c = f(N, \tau, \dot{v}_c, \rho_c)$

The engine speed will typically be measured by a tachometer, the engine torque by a magnetostrictive torque sensor, the flow by a positive displacement meter and the fuel temperature by a thermocouple. These instruments have inherent accuracy limitations and typically, the greater the accuracy the greater the procurement,

installation and maintenance costs. Therefore the calculation of the specific fuel consumption on an engine has an error that can be estimated by propagating the errors of the independent variables as measured by the instrumentation.

Propagation of error for several variables can be simply calculated in percentage units as follows (Weisstein, 2004):

For four variables, X, U, V, W, the function $Y = X \cdot U \cdot V \cdot W$ has a squared standard deviation in absolute units of

$$s_Y^2 = (UVW)^2 s_X^2 + (XVW)^2 s_U^2 + (XUW)^2 s_V^2 + (XUV)^2 s_W^2 \tag{A.3.4}$$

or

$$s_Y^2 = Y\left(\frac{s_X^2}{X^2} + \frac{s_U^2}{U^2} + \frac{s_V^2}{V^2} + \frac{s_W^2}{W^2}\right) \tag{A.3.5}$$

In percentage units, the standard deviation can be written as

$$\frac{s_Y}{Y} = \left(\frac{s_X^2}{X^2} + \frac{s_U^2}{U^2} + \frac{s_V^2}{V^2} + \frac{s_W^2}{W^2}\right)^{\frac{1}{2}} \tag{A.3.6}$$

Equation (A.3.6) assumes that all the covariances are negligible and that the function is a simple multiplicative function of the secondary variables. In this instance the assumptions are valid and therefore the propagated error for the diesel engine specific fuel consumption calculation is given by:

$$\frac{s_{Q_c}}{Q_c} = \left\{\left(\frac{s_N}{N}\right)^2 + \left(\frac{s_\tau}{\tau}\right)^2 + \left(\frac{s_{\dot{v}_c}}{\dot{v}_c}\right)^2 + \left(\frac{s_{\rho_c}}{T_c}\right)^2\right\}^{\frac{1}{2}} \tag{A.3.7}$$

The percentage errors accepted for the error propagation calculation are shown in Table A.3.1.

Table A.3.1: Instrumentation Error and Propagated Error for Specific Fuel Consumption

Measured Variable	Units	Instrument	BS 5514 ISO3046-3 Permissible Deviation	Acceptable Error	Propagated Error (%)
N	rev/s	Tachometer	±2%	**1%**	**1.40**
τ	N.m	Magnetostrictive torque sensor	±2%	**1%**	
$\dot{v}_c$	dm3/s	Positive displacement fuel meters (differential)	±3%	**1.5**	
ρ_c	kg/m^3	Calculation using thermocouple	±1.5% (±5 K on 320 K)	**0.75**	
Q_c **(calc)**	**g/kW.h**		**±3%**		**2.2**

In Table A.3.1, the permissible error (or deviation) of the primary measurements has been included from BS 5514 Part 3 as a guideline. The acceptable error for each of the primary measurements has been taken as 50% of the guideline permissible error.

The propagated error shown in Table A.3.1 is calculated using equation (A.3.1) together with equation (A.3.7). The results show that a goal for the neural network model error is justified to be set at ±2.5%. Further, from this goal the following error criteria are set:

Acceptable error: $\varepsilon_a = 0.025$

Mean propagated error of test data set: $\varepsilon_R = 0.025$

Model estimated mean propagated error: $\varepsilon_M = 0.01$

The error acceptance levels are to set the following:

- MSE error goal on training and test using the propagated error as the Nearer Specification Limit (NSL)
- AIC error on training and test using the MSE goal
- Test for means within the Nearer Specification Limit
- Confidence interval offsets for the black-box validation testing

A.3.2 Variable Perturbation Methods

The perturbation methods applied to the network sensitivity are described here. Two methods were applied, namely, the input-response mapping and the input to response sensitivity.

A.3.2.1 Variable Perturbation: Input-Response Mapping

The input variable response mapping method used in this study is defined in this section. The method was used to graph the input-response relationship or mapping of the neural networks. Its purpose was to verify whether each trained network was exhibiting the expected input to response behaviour as defined by known diesel engine operational performance characteristics, engine design and thermodynamic theory. In this study, the method is known as white-box validation. Neural network responses departing from the expected or known behaviour must be critically reviewed or retrained prior to final testing with the test data set.

The input-response mapping is executed as follows:

For each input variable and using the training data set

1. Calculate the mean

$$\overline{x}_R = \frac{1}{N}\sum_{i=1}^{N} x_{R_i} \tag{A.3.8}$$

where R is the variable number and n is the sample size (training set size).

2. Estimate the sample variance

$$s_R^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_{R_i} - \overline{x}_R)^2 \tag{A.3.9}$$

3. Estimate the sample standard deviation

$$S_R = \sqrt{s_R^2} \tag{A.3.10}$$

4. Define the perturbation lower and upper limits and range where

Perturbation Lower limit $$P_{R_0} = \overline{x}_R - S_R \quad \text{(A.3.11)}$$

Perturbation Upper limit $$P_{R_N} = \overline{x}_R + S_R \quad \text{(A.3.12)}$$

Perturbation Range $$P_R = 2S_R \quad \text{(A.3.13)}$$

5. Divide the perturbation range into n equal steps to create a perturbation increment

$$\delta P_R = \frac{2S_R}{n} \quad \text{(A.3.14)}$$

where n =100 in this research.

6. Create a new input column vector for the variable from the perturbation lower limit to the upper limit by the calculated perturbation increment, giving

$$\mathbf{x}_R = \left[P_{R_0} \; P_{R_1} \; P_{R_2} \; \ldots P_{R_n} \right]^{\mathrm{T}} \quad \text{(A.3.15)}$$

where $$P_{R_n} = P_{R_{n-1}} + \delta P_R$$

There now exists a new input data set for the variable.

7. To create a complete data set for the input to the trained network, create the data set with the perturbed variable keeping all the other input variables at their mean value. That is

$$\begin{aligned} \mathbf{X}_{1_n} &= \left[\mathbf{x}_{1_n} \; \overline{x}_2 \; \overline{x}_3 \; \overline{x}_4 \; \overline{x}_5 \; \overline{x}_6 \right] \\ \mathbf{X}_{2_n} &= \left[\overline{x}_1 \; \mathbf{x}_{2_n} \; \overline{x}_3 \; \overline{x}_4 \; \overline{x}_5 \; \overline{x}_6 \right] \\ &\vdots \\ \mathbf{X}_{6_n} &= \left[\overline{x}_1 \; \overline{x}_2 \; \overline{x}_3 \; \overline{x}_4 \; \overline{x}_5 \; \mathbf{x}_{6_n} \right] \end{aligned} \quad \text{(A.3.16)}$$

There now exists the perturbation test data set

$$\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}, \mathbf{X_4}, \mathbf{X_5}, \mathbf{X_6} \quad \text{(A.3.17)}$$

8. Apply each variable perturbation dataset to the trained network and measure the network response $\mathbf{y}_R$ for each variable perturbation data set.

9. Then graph the input variable $\mathbf{x}_R$ to its responses $\mathbf{y}_R$ to assist with verifying that the input-response mapping corresponds to the domain expert understanding of the relationship.

A.3.2.2 Variable Perturbation: Input Response Sensitivity

The second sensitivity analysis method applied for the neural network model verification calculates the ratio between the variance of a specific input variable to the variance of the network response to that variable; with all other variables held at their perturbation mean. Its purpose is to verify the scale of the model response sensitivity to the input when compared to the sensitivities to the other input variables. Typically, input variables with low sensitivity would be excluded from the neural network.

For each variable the sensitivity analysis method is as follows:

1. Calculate the variance of the perturbed input variable using the perturbation data set

$$s_R^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_{R_i} - \overline{x}_R)^2 \qquad \text{(A.3.18)}$$

2. Apply the variable perturbation dataset to the trained network and measure the network responses $\mathbf{y}_R$ for each variable perturbation data set.

3. Calculate the variance of the network response

$$s_{y_R}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{y}_{R_i} - \overline{y}_R)^2 \qquad \text{(A.3.19)}$$

4. Sensitivity is defined by calculating the ratio between the network response variance and the variance of the perturbed input

$$\mathrm{R}_{xy} = \frac{s_{y_R}^2}{s_R^2} \qquad \text{(A.3.20)}$$

The higher the ratio, the more sensitive the network response is to the specified input variable.

A.3.3 Paired, Independent and Single Sample t Tests

The paired and independent sample *t* tests are defined here together with the single sample t test. The procedures documented here have been adopted in this research when hypothesising the mean.

Considering the samples of two process models A and B together with the samples from a real process C, then:

- The paired sample *t* test is done when considering a single model and hypothesising the means within this model.
- The independent sample *t* test is done when hypothesising means between (across) the models or with the real process; since they are all independent processes.
- The single sample t test is done when comparing the model response to the real process at specified single valued contours on the engine performance map.

A.3.3.1 Independent Sample t-Test

Considering, for example, that there are two neural networks A and B and that n multiple runs are performed with A and B. Now letting $X_A = [X_{A_1}, X_{A_2}, \ldots, X_{A_n}]$ represent the n observed performance measures of network A and $X_B = [X_{B_1}, X_{B_2}, \ldots, X_{B_n}]$ represent the n observed performance measures of network B. The corresponding means of the multiple run performance measures are $\overline{X}_A$ and $\overline{X}_B$ with variances S_A^2 and S_B^2 for each network respectively.

To test the null hypothesis of no difference in the means of two independent populations

$$H_0: \quad \mu_A = \mu_B$$

against the alternative that the means are not the same

$$H_a: \quad \mu_A \neq \mu_B$$

the *t* value can be computed using the pooled-variance *t* test (Weisstein, 2004).

$$t_{n_A+n_B-2} = \frac{\left(\bar{X}_A - \bar{X}_B\right) - \left(\mu_A - \mu_B\right)}{\sqrt{S_p^2\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}} \tag{A.3.21}$$

with $n_A + n_B - 2$ degrees of freedom and where the pooled variance

$$S_p^2 = \frac{\left(n_A - 1\right)S_A^2 + \left(n_B - 1\right)S_B^2}{\left(n_A - 1\right) + \left(n_B - 1\right)} \tag{A.3.22}$$

The same test is performed when, for example, the mean response of a neural network or quadratic model is compared to the test data set from an actual engine.

The pooled-variance *t* test is conducted under the assumptions of normally distributed sample populations and equal variances. If there is evidence that the variances are not equal, then the Behrens-Fisher problem exists and the pooled-variance *t* test is inappropriate. The Satterthwaite separate-variance *t'* test is then to be employed as

$$t_\upsilon' = \frac{\left(\bar{X}_A - \bar{X}_B\right) - \left(\mu_A - \mu_B\right)}{\sqrt{\frac{S_A^2}{n_A} + \frac{S_B^2}{n_B}}} \tag{A.3.23}$$

with the degrees of freedom taken as the integer portion of

$$\upsilon = \frac{\left(\frac{S_A^2}{n_A} + \frac{S_B^2}{n_B}\right)^2}{\frac{\left(\frac{S_A^2}{n_A}\right)^2}{n_A - 1} + \frac{\left(\frac{S_B^2}{n_B}\right)^2}{n_B - 1}} \tag{A.3.24}$$

A.3.3.2 Paired or Related Sample t Test

Given the two paired sets X_{A1} and X_{A2} of n measured values from the same model with corresponding means of $\overline{X}_{A1}$ and $\overline{X}_{A2}$ respectively, the paired *t* test determines whether they differ from each other in a significant way under the assumptions that the paired differences are independent and identically normally distributed (Weisstein, 2004).

To apply the test, let

$$\hat{X}_{A1_i} = (X_{A1_i} - \overline{X}_{A1}) \tag{A.3.25}$$

$$\hat{X}_{A2_i} = (X_{A2_i} - \overline{X}_{A2}) \tag{A.3.26}$$

then t is define by

$$t_{n-1} = (\overline{X}_{A1} - \overline{X}_{A2}) \sqrt{\frac{n(n-1)}{\sum_{i=1}^{n} (\hat{X}_{A1_i} - \hat{X}_{A2_i})^2}} \tag{A.3.27}$$

with $n-1$ degrees of freedom.

A.3.3.3 Single Sample t Test

When comparing the neural network model response mean to the target mean at the constant specific fuel consumption lines (contours), one is required to conduct a single sample *t* test. This is because the standard deviation of the target mean is unknown at the contours.

Considering that the neural network response to selected inputs is X_{Ai} and the target mean for the map contour is $\overline{\mu}_c$, then the *t* value of the single sample *t* test can be defined as

$$t_{n-1} = \frac{\overline{X}_A - \overline{\mu}_c}{\frac{S_A}{\sqrt{n}}} \tag{A.3.28}$$

where $\overline{X}_A$ is the sample mean of the network response for n number of input exemplars

$$\overline{X}_A = \frac{1}{n}\sum_{i=1}^{n} X_{A_i} \tag{A.3.29}$$

and the sample variance is given by

$$S_A = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_{A_i} - \overline{X}_A)^2} \tag{A.3.30}$$

A.3.4 Linear Regression (Least Squares Method)

Linear regression is not used in this study for input-response mapping, rather it is used to compare the model response to the target response.

The sample regression equation representing the linear regression model is given by

$$\hat{y}_i = b_0 + b_1 x_i \tag{A.3.31}$$

where: $\hat{y}_i$ is the predicted value of y_i for observation x_i and the two parameters b_0 and b_1 correspond to the y-intercept and gradient of the best linear regression relating targets to model outputs.

The model coefficients are determined using the normal equations given by

$$\sum_{i=1}^{n} y_i = nb_0 + b_1 \sum_{i=1}^{n} x_i \tag{A.3.32}$$

$$\sum_{i=1}^{n} x_i y_i = b_0 \sum_{i=1}^{n} x_i + b_1 \sum_{i=1}^{n} x_i^2 \tag{A.3.33}$$

and solving simultaneously, gives

$$b_1 = \frac{\sum_{i=1}^{n} x_i y_i - n\,\overline{x}\,\overline{y}}{\sum_{i=1}^{n} x_i^2 - n\,\overline{x}} = \frac{ss_{xy}}{ss_{xx}} \tag{A.3.34}$$

$$b_0 = \overline{y} - b_1 \overline{x} \tag{A.3.35}$$

A.3.5 Change Point Analysis

Did a change occur? Did more than one change occur? When did the changes occur? With what confidence did the changes occur? All these questions and more can be answered by performing a change-point analysis.

A change-point analysis (CPA) is capable of detecting multiple changes. For each change it provides detailed information including a confidence level indicating the likelihood that a change occurred and a confidence interval indicating when the change occurred. The CPA procedure is extremely flexible. It can be performed on all types of time ordered data including attribute data, data from non-normal distributions, ill-behaved data such as particle counts and complaint data, data with outliers and specifically in this demonstration the fuel consumption variance data.

CPA is a powerful tool for determining whether a change has taken place and when it took place, for example, has the specific fuel consumption changed and in which time period? Understanding the time of change is important in industrial applications, since complimentary process data can be reviewed to enhance the understanding of why a change occurred.

CPA is primarily used for historical data (which is precisely the approach taken in energy management). When analyzing historical data, especially when dealing with large data sets, change-point analysis is preferable to control charting. CPA is more powerful, better characterizes the changes, controls the overall error rate, is robust to outliers, is more flexible and is simpler to use.

In the sections that follow CPA is defined in terms of the engine performance mapping demonstration and how the neural network model compliments the process.

A.3.5.1 Procedure for Performing a Change-Point Analysis

The procedure used by Taylor (2000) for performing a change-point analysis iteratively uses a combination of Cumulative Sum Charts (CUSUM) and bootstrapping to detect the changes. The analysis begins with the construction of the

CUSUM chart. CUSUM charts are constructed by calculating and plotting a cumulative sum based on the data. The method is processed as follows:

Let $\Delta M_c(1), \Delta M_c(2), \ldots, \Delta M_c(n)$ represent the n calculated fuel consumption performance variances from target as defined by equation (7.1). Note that these variances are in time series order. From this, the cumulative sums $S(0), S(1), \ldots, S(n)$ can be calculated as follows:

- First calculate the average

$$\Delta \bar{M}_c = \frac{\Delta M_c(1) + \Delta M_c(2) + \ldots + \Delta M_c(n)}{n} \quad \text{(A.3.36)}$$

- Start the cumulative sum at zero by setting

$$S(0) = 0$$

- Calculate the other cumulative sums by adding the difference between current value and the average to the previous sum, that is

$$S(i) = S(i-1) + (\Delta M_c(i) - \Delta \bar{M}_c) \quad \text{(A.3.37)}$$

for $i = 1, 2, ..., n$

The CUSUM values are then graphed. However, note that the cumulative sums are not the cumulative sums of the values. Instead they are the cumulative sums of differences between the values and the average. These differences sum to zero so the cumulative sum always ends at zero (see BS5703-1/2/3 for CUSUM construction and interpretation). Changes in gradient can indicate change points (a shift in the mean of the difference). The problem with CUSUM charts is that they require considerable skill to properly interpret. Therefore what level of confidence can be placed on an apparent change? This is a serious question, since intervention (for example maintenance or further investigation) is expensive and disruptive in an industrial setting. Thus, one needs to apply a technique for determining the confidence level of a change.

A confidence level can be determined for the apparent change by performing a bootstrap analysis (Efron, 1993). However, prior to performing the bootstrap analysis, an estimator of the magnitude of the change is required. One option for the estimator which works well, regardless of the distribution and multiple change points, is provided by Taylor (2000)

$$S_{\text{diff}} = S_{\max} - S_{\min} \tag{A.3.38}$$

where $S_{\max}$ is the maximum value and $S_{\min}$ the minimum value of $S(i)$ respectively for $i = 0,1,\ldots,n$

Once the estimator of the magnitude of the change has been determined, the bootstrap analysis can be performed. A single cycle of the bootstrap is performed as by executing the following algorithm:

- Generate a bootstrap sample of n fuel consumption performance variances denoted as

$$\Delta M_c^0(1), \Delta M_c^0(2), \ldots, \Delta M_c^0(n) \tag{A.3.39}$$

  by randomly reordering the original n values; which is a basic process of sampling without replacement.

- As before and based on the bootstrap sample, calculate the bootstrap CUSUM, denoted as

$$S^0(0), S^0(1), \ldots, S^0(n) \tag{A.3.40}$$

- Calculate the maximum, minimum and difference of the bootstrap CUSUM to obtain the magnitude estimator for the bootstrap sample, that is

$$S_{\text{diff}}^0 = S_{\max}^0 - S_{\min}^0 \tag{A.3.41}$$

- Determine whether the bootstrap magnitude estimator S_{diff}^0 is less than the original S_{diff}

The bootstrap analysis consists of performing a large number of bootstrap cycles and counting the number of bootstraps for which $S_{\text{diff}}^0 < S_{\text{diff}}$. Therefore letting N be the

number of bootstrap cycles performed and let X be the number of bootstraps for which $S_{\text{diff}}^{0} < S_{\text{diff}}$, then the confidence level α_{change} that a change occurred as a percentage is calculated as follows:

$$\alpha_{\text{change}} = 100\frac{X}{N}\% \tag{A.3.42}$$

The number of bootstraps N is typically 1000 which is sufficient for most purposed and a 90% or 95% confidence level is required before a statement can be made that a significant change in performance has been detected. There are numerous other statistics that can be computed for CPA, for example, the confidence interval of the change point, test for change in variance and control limits. These are not detailed or discussed here since they do not support further understanding of the demonstration.

A.3.6 Box and Whisker Plots – Definition

Box and Whisker plots (box plots) are helpful in data analysis for interpreting the distribution of data since it can easily show whether the data is skewed and if there are unusual observations (outliers) in the dataset. Box plots are also very useful when large numbers of observations are involved and when two or more datasets are being compared. Figure A.3.1 shows a typical box plot. In this study the box is plotted in a vertical orientation.

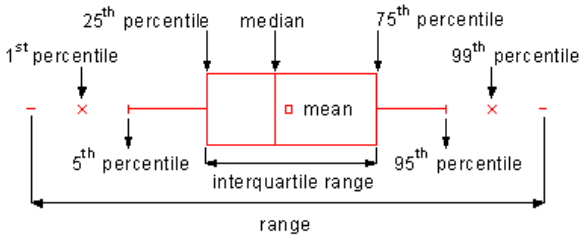


Figure A.3.1: Box Plot Definition

The Box plot (Tukey, 1977) displays a statistical summary of a variable: median, quartiles, range and, possibly, extreme values. In the Box plot, the central box represents the values from the lower to upper quartile (25 to 75 percentile). The middle line represents the median. The horizontal line extends from the minimum to the maximum value, excluding outside and far out values which can be displayed as separate points.

An outside value (outlier) is defined as a value that is smaller than the lower quartile minus 1.5 times the interquartile range, or larger than the upper quartile plus 1.5 times the interquartile range (inner fences).

Appendix A.4

Tabulated Results Statistics

A.4.1 Neural Network Performance – Mean Squared Error on Multiple Training

Table A.4.1: Model Performance Error (ε_{ms}) – Multiple-Run Training Descriptive Statistics

N=20			F-mse-tr	F-mse-cv	M-mse-tr	M-mse-cv
Statistic	Mean		14.18	17.38	18.03	22.12
	95% Confidence Interval for Mean	Lower Bound	9.44	11.11	16.32	20.03
		Upper Bound	18.92	23.66	19.75	24.21
	5% Trimmed Mean		12.28	14.79	18.10	22.21
	Median		11.53	14.00	18.13	22.15
	Variance		102.64	179.63	13.42	19.95
	Std. Deviation		10.13	13.40	3.66	4.47
	Minimum		7.97	9.92	11.64	14.23
	Maximum		54.58	71.54	23.11	28.36
	Range		46.61	61.62	11.47	14.13
	Interquartile Range		5.62	6.18	6.68	8.13
	Skewness		3.66	3.81	-.20	-.24
	Kurtosis		14.79	15.76	-1.09	-1.06
Std. Error	Mean		2.27	3.00	.82	1.00
	Skewness		.51	.51.	51	.51
	Kurtosis		.99	.99	.99	.99

A.4.2 Neural Network Performance – AIC on Multiple Training

Table A.4.2: Model Performance Error (AIC) – Descriptive Statistics

	N=20		F-aic-tr	F_aic_cv	M_aic_tr	M_aic_cv
Statistic	Mean		23282	24989	25730	27542
	95% Confidence Interval for Mean	Lower Bound	21456	23121	24845	26659
		Upper Bound	25109	26857	26616	28424
	5% Trimmed Mean		22781	24437	25803	27616
	Median		22544	24260	25956	27732
	Variance		15226361	15934075	3580087	3555670
	Std. Deviation		3902	3992	1892	1886
	Minimum		19279	21209	22038	23820
	Maximum		36308	38702	28113	29926
	Range		17028	17493	6074	6106
	Interquartile Range		4275	4029	3320	3312
	Skewness		2.063	2.275	-.508	-.542
	Kurtosis		5.895	6.923	-.712	-.658
Std. Error	Mean		873	893	423	422
	Skewness		.512	.512	.512	.512
	Kurtosis		.992	.992	.992	.992

A.4.3 Model Response Mean and Target Mean – Test Data Set

Table A.4.3: Model Response Mean to Target Mean – Descriptive Statistics

	N=699		T-target	F-response	M-response	G-response
Statistic	Mean		236.51	242.34	238.83	225.89
	95% Confidence Interval for Mean	Lower Bound	235.32 *230.59	241.19	237.91	225.24
	*adjusted	Upper Bound	237.69 *242.42	243.49	239.74	226.54
	5% Trimmed Mean		235.91	241.65	238.51	225.63
	Median		230.00	237.65	236.10	225.28
	Variance		255.04	239.01	151.13	76.36
	Std. Deviation		15.97	15.46	12.29	8.74
	Minimum		212.00	217.64	217.57	211.89
	Maximum		270.00	278.32	264.80	246.16
	Range		58.00	60.68	47.23	34.26
	Interquartile Range		25.00	23.62	21.06	15.35
	Skewness		.58	.61	.39	.27
	Kurtosis		-.73	-.50	-.85	-.84
Std. Error	Mean		.60	.58	.46	.33
	Skewness		.09	.09	.09	.09
	Kurtosis		.18	.18	.18	.18

A.4.4 Model Response Mean for Target Contour Mean

Table A.4.4: Model Response Means at Target Contours ($\varepsilon_M = 0.01$)

			Contour Value 212 (n=27)			Contour Value 220 (n=137)		
			F-resp	M-resp	G-resp	F-resp	M-resp	G-resp
Statistic	Mean (g/kW.h)		221.31	221.21	219.67	225.74	225.17	223.32
	95% Confidence Interval for Mean (adjusted for ε_M)	Lower Bound	221.04 (218.8)	221.00 (218.8)	219.00 (216.8)	225.16 (222.9)	224.57 (222.3)	221.64 (219.4)
		Upper Bound	221.58 (223.8)	221.42 (223.6)	220.34 (222.5)	226.32 (228.6)	225.76 (228.0)	224.99 (227.2)
	Median		221.32	221.20	219.95	226.60	226.12	219.80
	Variance		0.48	0.27	2.86	11.83	12.37	98.65
	Std. Deviation		0.69	0.52	1.69	3.44	3.52	9.93
	Minimum		220.32	220.50	217.22	217.64	217.57	211.89
	Maximum		222.28	221.96	221.87	229.27	229.21	238.95
	Range		1.96	1.46	4.66	11.63	11.64	27.06
	Interquartile Range		1.32	1.07	3.39	4.33	5.53	21.35
	Skewness		-0.02	0.00	-0.17	-0.92	-0.78	0.36
	Kurtosis		-1.49	-1.54	-1.59	-0.23	-0.64	-1.55
Std. Error	Mean		0.13	0.10	0.33	0.29	0.30	0.85
	Skewness		0.45	0.45	0.45	0.21	0.21	0.21
	Kurtosis		0.87	0.87	0.87	0.41	0.41	0.41

			Contour Value 225 (n=95)			Contour Value 230 (n=114)		
			F-resp	M-res	G-resp	F-resp	M-resp	G-resp
Statistic	Mean		232.11	231.17	219.32	235.95	234.08	221.47
	95% Confidence Interval for Mean (adjusted for ε_M)	Lower Bound	231.86 (229.5)	230.76 (228.5)	218.00 (215.8)	235.66 (223.3)	233.64 (231.3)	220.32 (218.1)
		Upper Bound	232.36 (234.7)	231.57 (223.9)	220.63 (222.8)	236.25 (238.6)	234.53 (236.9)	222.63 (224.8)
	Median		232.49	231.82	217.29	236.18	235.02	219.61
	Variance		1.54	3.95	41.62	2.52	5.64	38.91
	Std. Deviation		1.24	1.99	6.45	1.59	2.38	6.24
	Minimum		230.02	227.88	212.90	233.38	229.70	214.87
	Maximum		233.54	233.47	237.20	237.95	236.85	236.81
	Range		3.52	5.59	24.29	4.57	7.15	21.94
	Interquartile Range		2.47	4.01	9.14	3.09	4.74	9.95
	Skewness		-0.45	-0.42	1.12	-0.25	-0.47	0.82
	Kurtosis		-1.37	-1.43	0.43	-1.48	-1.36	-0.41
Std. Error	Mean		0.13	0.20	0.66	0.15	0.22	0.58
	Skewness		0.25	0.25	0.25	0.23	0.23	0.23
	Kurtosis		0.49	0.49	0.49	0.45	0.45	0.45

			Contour Value 240 (n=128)			Contour Value 250 (n=78)		
			F-resp	M-resp	G-resp	F-resp	M-resp	G-resp
Statistic	Mean		245.80	241.74	225.24	255.04	250.60	229.36
	95% Confidence Interval for Mean (adjusted for ε_M)	Lower Bound	245.48 (243.0)	241.19 (238.8)	224.57 (222.3)	254.62 (252.1)	250.26 (247.8)	229.03 (226.7)
		Upper Bound	246.12 (248.6)	242.30 (244.7)	225.92 (228.2)	255.46 (258.0)	250.93 (253.4)	229.70 (232.0)
	Median		246.12	242.67	224.05	255.17	250.97	228.75
	Variance		3.42	10.01	14.90	3.42	2.19	2.20
	Std. Deviation		1.85	3.16	3.86	1.85	1.48	1.48
	Minimum		242.79	235.27	220.99	251.81	247.19	227.98
	Maximum		248.20	244.89	235.18	257.34	252.22	233.43
	Range		5.41	9.62	14.19	5.53	5.03	5.46
	Interquartile Range		3.83	5.54	6.32	3.49	2.46	1.88
	Skewness		-0.23	-0.68	0.79	-0.29	-0.80	1.24
	Kurtosis		-1.49	-0.96	-0.51	-1.33	-0.55	0.48
Std. Error	Mean		0.16	0.28	0.34	0.21	0.17	0.17
	Skewness		0.21	0.21	0.21	0.27	0.27	0.27
	Kurtosis		0.42	0.42	0.42	0.54	0.54	0.54

			Contour Value 260 (n=76)			Contour Value 270 (n=44)		
			F-resp	M-resp	G-resp	F-resp	M-resp	G-resp
Statistic	Mean		263.40	255.18	234.83	276.63	263.42	243.69
	95% Confidence Interval for Mean (adjusted for ε_M)	Lower Bound	263.11 (260.5)	254.67 (252.1)	234.56 (232.2)	276.17 (273.4)	263.10 (260.5)	243.36 (240.9)
		Upper Bound	263.70 (266.3)	255.70 (258.3)	235.09 (237.4)	277.10 (279.9)	263.74 (266.4)	244.03 (246.5)
	Median		264.27	255.66	234.39	277.13	263.47	243.31
	Variance		1.67	5.10	1.31	2.31	1.11	1.21
	Std. Deviation		1.29	2.26	1.14	1.52	1.05	1.10
	Minimum		261.00	250.94	233.61	273.17	261.36	242.36
	Maximum		264.80	258.60	237.84	278.32	264.80	246.16
	Range		3.80	7.66	4.23	5.15	3.44	3.80
	Interquartile Range		2.35	3.74	1.66	2.36	2.01	1.93
	Skewness		-0.70	-0.31	0.98	-0.85	-0.40	0.56
	Kurtosis		-1.15	-1.10	-0.12	-0.58	-1.00	-0.93
Std. Error	Mean		0.15	0.26	0.13	0.23	0.16	0.17
	Skewness		0.28	0.28	0.28	0.36	0.36	0.36
	Kurtosis		0.54	0.54	0.54	0.70	0.70	0.70

[Blank Page]