Transient Engine Model for Calibration Using Two-Stage Regression Approach

Muhammad Alam Zaib Khan

Aeronautical and Automotive Engineering

Loughborough University

Submitted in partial fulfilment of the requirements for the award of

PhilosophiæDoctor (PhD)

December, 2010

Abstract

Engine mapping is the process of empirically modelling engine behaviour as a function of adjustable engine parameters, predicting the output of the engine. The aim is to calibrate the electronic engine controller to meet decreasing emission requirements and increasing fuel economy demands. Modern engines have an increasing number of control parameters that are having a dramatic impact on time and effort required to obtain optimal engine calibrations. These are further complicated due to transient engine operating mode.

A new model-based transient calibration method has been built on the application of hierarchical statistical modelling methods, and analysis of repeated experiments for the application of engine mapping. The methodology is based on two-stage regression approach, which organise the engine data for the mapping process in sweeps. The introduction of time-dependent covariates in the hierarchy of the modelling led to the development of a new approach for the problem of transient engine calibration.

This new approach for transient engine modelling is analysed using a small designed data set for a throttle body inferred air flow phenomenon. The data collection for the model was performed on a transient engine test bed as a part of this work, with sophisticated software and hardware installed on it. Models and their associated experimental design protocols have been identified that permits the models capable of accurately predicting the desired response features over the whole region of operability.

Further, during the course of the work, the utility of multi-layer perceptron (MLP) neural network based model for the multi-covariate case has been demonstrated. The MLP neural network performs slightly better than the radial basis function (RBF) model. The basis of this comparison is made on assessing relevant model selection criteria, as well as internal and external validation fits.

Finally, the general ability of the model was demonstrated through the implementation of this methodology for use in the calibration process, for populating the electronic engine control module lookup tables.

Index Terms -- Engine Mapping, Model Based Calibration, Two-Stage Regression, Transient Engine Model, Transient Engine Calibration, Hierarchical Models, Non-Linear Repeated Measurements, Multi-Layer Perceptron, Radial Basis Functions, Transient Air Flow Model

The thesis is dedicated to my parents, who taught me the value and importance of education. And also, special dedication to my wife and my kids, who offered me unconditional love and support throughout the course of this thesis.

Acknowledgements

I would like to thank Dr. Rui Chen of the Loughborough University. I am very grateful for both his supervision and help over the last few years.

I also acknowledge Higher Education Commission (HEC), Islamabad for their financial assistance.

Table of Contents

List of Figures

List of Tables

Glossary

Abbreviations

ΔLESS10 Spark Timing Retarded 10 degree from MBT.

ΔLESS25 Spark Timing Retarded 25 degree from MBT.

ΔPLUS Spark Timing Advanced 10 degree from MBT.

AFR Air-Fuel-Ratio.

AIC Akaike's Information Criterion.

AIC$_C$ Small Sample version of AIC.

AN Artificial Neuron.

BDC Bottom Dead Center.

BIC Bayesian Information Criterion.

BTQ Brake Torque.

CAGE Calibration Generation.

CAN Control Area Network.

CCD Central Composite Design.

CCRD Central Composite Rotatable Design.

CVIVL Continuously Variable Intake Valve Lift.

DEPE Design of Experiments for Powertrain Engineering Consortium.

DoE Design of Experiments.

DOHC Dual Over Head Cam.

ECP Exhaust Cam Phase.

ECU Electronic Control Unit.

EMS Engine Management System.

EXTEMP Exhaust Temperature.

FFD Full Factorial Design.

GCV Generalised Cross Validation.

HBSP Hybrid B-Spline Polynomial.

ICP Intake Cam Phase.

IMEP Indicated Mean Effective Pressure.

LHS Latin Hypercube Sampling.

LOAD Normalised Induced Air Charge.

LOO Leave One Out cross validation.

MAF Mass Air Flow.

MAP Manifold Pressure.

MARS Multivariate Adaptive Regression Splines.

MBC Model Based Calibration.

MBT Spark angle that generate Maximum Brake Torque, p. xii.

MLP Multi-Layer Perceptron.

MSE Mean Squared Error.

N Engine Speed.

NN Neural Network.

PEV Prediction Error Variance.

PFI Port Fuel Injection.

PKTQ Peak Torque.

RBF Radial Basis Function.

RFRAC Residual Fraction.

RMSE Root Mean Square Error.

RSM Response Surface Method.

S Spark Advance.

STP Standard Temperature and Pressure.

Nomenclature

β_i Vector of level-1 regression parameters.

Γ Level-2 covariance matrix.

γ Vector of level-2 conditionally linear fixed effect parameters.

σ^2 Pooled level-1 coefficient of variation.

θ Vector of level-2 fixed effects.

ζ Vector of pooled level-1 dispersion model parameters.

a_i Vector of level-2 covariates.

b_i Vector of level-2 random effects.

$d(a_i, \theta, b_i)$ Vector valued function describing level-2 systematic and random variation.

e_i Intra-sweep error vector for the i^{th} sweep.

e_{ij} j^{th} observed intra-sweep error for the i^{th} sweep.

$f(x_{ij}, \beta_i)$ Level-1 fit function.

g_i Vector of response features for the i^{th} sweep.

m number of sweep.

n_i number of observations for the i^{th} sweep.

p Dimension of level-1 regression parameter or response feature vectors.

q Number of level-1 covariance model parameters.

r Dimension of level-2 fixed effects vector.

$v^2(\beta_i, \zeta_i)$ Function describing the level-1 variance heterogeneity.

x_{ij} j^{th} level-1 covariate vector for the i^{th} sweep.

y_i Vector of observed responses for the i^{th} sweep.

y_{ij} j^{th} observed response for the i^{th} sweep.

Notation

N Denotes the normal distribution.

$x \in R^n$ x is an n-by-1 column vector of real elements.

$X \in R^{r \times c}$ The set of real numbers.

E Expectation operator, $E[x] = \int_{-\infty}^{\infty} xp(x)dx$, with $p(x)$ the p.d.f for x.

R The set of real numbers.

Var Variance operator, $Var[x] = \int_{-\infty}^{\infty}(x - E[x])^2 p(x)dx$, with $p(x)$ the p.d.f for x.

1

Introduction

1.1 Introduction

Today, modern engine systems are equipped with complex technologies such as multiple injections, exhaust gas recirculation, combustion and after-treatment systems; in response to the decreasing emission requirements and increasing fuel economy demands. As a result, the challenges for engine calibration and control are increasing. Further, even as the requirements of complying with ever lower transient emissions regulations cannot be underestimated. The conventional engine calibration techniques are time consuming, adhoc and repetitive, resulting in low productivity of test facilities and engineering effort. These techniques will be unable to keep up with the increased demands in workload and accuracy required in future highly complex engine and after-treatment systems. As a result of this, the calibration would be impossible to perform entirely in the test cell on these complex engines.

There is a growing realisation of reducing development cost by moving as much of the engine calibration process out of the engine test cell onto the desktop environment, using model-based calibration methods. This method offers significant advantages which helps in the reduction of time and effort for optimised engine calibration.

There have been significant advances made in steady-state engine calibration processes such as adaptive online data acquisition (5), rule based calibration (6) and simulation based calibration (7; 8). Response Surface Method (RSMs) (9; 10) is most commonly used method for steady-state engine calibration which typically use Design of Experiments (DoE) to obtain data using a structured test plan and then typically use regression models to study the relationship between the independent and response variables. These models then drive the DoEs towards further data acquisition or final engine calibrations. Other examples of practical applications of RSMs for engine system and engine subsystem calibration can be found in the work of Montgomery and Reitz (11; 12), Edwards and Pilley (13), Burk et al. (14) and Dimopoulos et al. (15). Examples of work where distinct engine operating points have been optimized individually or have been considered as a group to meet some overall cycle emission limits, while maintaining mechanical and thermal limits at each engine operating point is included in work by Kampelmuhler *et al* (16) who used second order models at each engine operating point to develop an automated calibration process. Schmitz *et al* (17) used a Lagrangian function for multi-point optimization as part of an calibration process development, Brooks *et al* (18) have used the Matlab Model Based Calibration (MBC) toolbox to develop calibrations to comply with the New European Drive Cycle and, Knafl *et al* (19) have used second order models to develop dual-use engine calibrations. Traditional RSM methods have used least square linear regression models including linear, pure quadratic and full quadratic models, but some recent modelling techniques have demonstrated significant promise and advantages over traditional RSMs.

Neural network models used for engine optimization such as (20; 21) can be used to represent the entire dataset. A structured test plan or DOEs are not usually required and any amount of non-linearity can be handled. A lot of recent work (22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34) has gone into utilizing the power of neural networks for engine modelling. Qiang et al. (35) have used multiple neural networks trained at distinct engine operating points and used fuzzy logic to interpolate at intermediate points for the purpose of engine calibration.

Other recently used techniques include Kriging techniques, Radial Basis Functions, Splines and Support Vector machines. Guerrier et al. (36) have examined Linear Growth Models and Radial Basis Function Models for engine calibration. The phenomenological models such as the soot model developed by Bayer and Foster (37; 38) and Bayesian techniques (39; 40) show significant promise in engine calibration development. The Design of Experiments for Powertrain Engineering Consortium (DEPE) led by Ricardo Inc. have claimed that their Stochastic Modeling Methods have outperformed polynomials, neural networks and Radial Basis Functions (41).

Most of these model based calibration process have focused on the steady state operation. These steady state engine mapping methods, such as design of experiments, do little to ensure transient emissions compliance or fuel consumption optimization, which implies that the engine calibration optimization, if it is done in an offline or 'virtual' environment, must be performed using a transient or dynamic engine model.

However, Atkinson et al. (42; 43; 44) used neural networks to predict transient engine operation. They used a hybrid equation-based and neural network-based data-driven technique to produce an engine model for calibration and optimisation. Brahma et al. (45) have used empirical modelling for transient emissions and response for optimisation using full quadratic global regression model. The model based has been used by Hafner (46) for determination of dynamic engine control parameters. Hafner (46) and Dohmen (47) work discuss transient engine testing procedure and post-processing techniques for transient data collection.

Transient engine calibration must account for *time* as an additional dimension. Therefore, transient modelling is different from the steady state modelling in various ways. The most important is the data acquisition and processing in transient condition is highly complex and is very important in view of model development. Also, the model developed for the steady state condition might not be suitable for the transient data.

1.2 Research Problem

From the calibration point view, there is a growing demand to meet the challenges faced by the increase in engine technologies, hence the extremely complex non-linear processes sensitive to large number of interacting factors. Also, the steady state engine mapping or calibration optimisation are not suited to the situation in which the prevailing emissions standard is transient standard, such as EPA Heavy Duty Transient Cycle (HDTTC), EPA Smoke Test, EURO III – Load Response Test and the FTP – 75 test for light-duty engine. In these case, the legislative requirement is based on the engine performance over a dynamic cycle, which implies that the engine calibration optimisation must be performed using transient or dynamic engine model (30). Transient engine model should take into account the full dynamic nature of an engine performance, in order to more accurately model the transient time-varying nature of the engine's emissions, performance and fuel consumption behaviour.

1.3 Research Objectives

This research has been initiated from a general observation that, despite the potential advantages of model-based methods in engine mapping process, the calibration is still limited to steady state engine conditions. The research presented in this thesis investigates a new approach to the development and implementation of a model-based transient engine model that can enable calibration in the transient condition with an added domain of *time* in the hierarchy of the model.

This research has been conducted with respect to the limited provision of transient calibration methods currently available in commercial and public domain. First, the emergence of new legislative requirements regarding engine performance over a dynamic cycle imposes new requirement regarding the design and implementation of calibration optimisation methods. Secondly, the development of such models taking into account the full dynamic nature of an engine performance imposes new constraints on the data acquisition and hence, implantation of these models.

The calibration process commences with a limited data collected in a transient test cell and is then transferred to the computational environment where dynamically predictive engine models are created. To prove the accuracy of the method, a number of transient data sets were collected from the same engine, are then validated on the developed engine model. The engine calibration is then generated off-line using these transient models. The following objectives were satisfied in order to achieve the research aim;

1. To identify hierarchical non-linear models with good predictive capability over the entire region of operability accounting *time* as an additional degree of freedom.
2. To determine appropriate design procedures to support the models.
3. To demonstrate the application of these models for engine calibration.

1.4 Software Tools

In order to develop the transient engine model, different software and tools were used during the course of studies. The details are as follows:

1.4.1 Test and Measurement Software

The Engine is remotely controlled by the controlling PC located in the control room, next to the test stand room via an Electronic Control Unit (ECU). An **ATI VISION** software is used to control and measure the data from the engine. ATI VISION Software is an integrated calibration and data acquisition tool that collects signals from the ECU and external sources, measures relationships between inputs and outputs, enables real-time calibration and modification to closed loop control systems, time aligns and analyses all information, manages calibration data changes and programs the ECU. VISION includes the following features:

- Recording of measurement data, calibration variables, and virtual data items

- Number of channels ($10 \sim 15$) and multiple channel sampling rates supported
- Multiple ($2 \sim 3$) measurement recorders can run simultaneously
- Multiple ($2 \sim 3$) trigger conditions
- Storage in ASCII, MATLAB ®
- Offline calibration without an ECU

Also, a custom made National Instruments (NI) **LabView** software was designed and build to allow transient testing of the powertrain under repeatable conditions in a laboratory environments, and to allow a comprehensive control of the dynamometer controller behaviour. The software can;

- Trigger the dynamometer
- Export the designs experiment table to the controller
- Record the dynamometer data

1.4.2 Analysis Software's

Much of the analysis for the research work is performed in the *Model Based Calibration Toolbox* (MBC) implemented in Matlab ®. MBC is a commercial toolbox jointly developed by *The Mathworks* and *Ford Motor Company*.

MBC provides design tools for optimally calibrating complex powertrain systems using statistical modelling and numeric optimization. It contains tools for design of experiment, statistical modelling, and for generating calibrations and lookup tables for complex high-degree-of-freedom engines. These features in toolboxes accomplished via two main user interfaces:

1. A **Model Browser** for experimental design and statistical modelling

- Designing a test plan based on Design of Experiments, a methodology that saves test time by letting to perform only those tests that are needed to determine the shape of the engine response. The toolbox offers a full range of proven experimental designs, including: Classical, space-filling, and optimal designs for creating optimized test plans
- Estimation of local and global variations separately by fitting local and global models in two stages. Two-stage modelling is used to map the complex relationships among all the variables that control the behaviour of the engine. An extensive range of built-in and user-definable libraries of empirical model forms is available at either level in the hierarchy. The necessary transient engine model specified in this thesis can be implemented directly in the toolbox.
- Boundary modelling to keep optimisation results within the engine operating envelope. Acquiring data and modelling the engine must account for the operating regions of the system that can be physically tested. MBC lets you add constraints to your experimental designs and create boundary models that describe the feasible region for testing and simulation.
- Evaluation and comparison of single and two-stage models for model selection. Different model performance measures, such as RMSE, PRESS and GCV is automatically calculated.
- Response feature visualisation tools for two- and three- dimensional surface projections, parametric plots and contour plots.

2. A **Calibration Generation (CAGE) Browser** for generation of lookup tables from models, optimization results, or test data

   - Generate optimal calibrations directly from empirical engine models
   - Producing smooth calibration tables by table-filling wizard that enables incremental filling of tables from the results of multiple optimisations with smooth interpolation through existing table values. The CAGE tool extrapolates the optimisation results to pass smoothly through table masks and locked cells (fixed table values).

- Compare calibrations with test data
- Export calibrations to ETAS INCA and ATI Vision

Hence, the toolbox provides an ideal environment for the design and analysis of the engine mapping experiments and subsequent application of the resultant models for analytical calibration purposes.

1.5 Thesis Organization

The thesis begins with the description of the calibration process in detail and discusses the current approach to engine mapping process, before presenting the challenge faced for the calibration due to new engine technologies. The need for model-based transient engine calibration is discussed in Chapter 2. Chapter 3 review and analyse model based methods with the discussion of different experimental design and, approximation models and fitting techniques. The chapter also covers fundamentals of the two-stage regression approach applied in engine mapping process and its mathematics. Recommendation of model choice and fitting is also presented.

Chapter 4 discusses the empirical model building such as experimental design and response feature methodology for the two-stage regression model using neural network. The experimental design is based on space filling approach, in which the available points are spread in a relatively uniform fashion on entire region to capture as much information as possible, and does not assume a particular model form. A space-filling design is best for exploring a new system where prior knowledge about the underlying effects of factors and responses is low.

Chapter 5 presents a steady state neural network based approach to the stage–2 modelling for engine calibration. The model is developed with good generalisation capability and interpolation accuracy. For the approximation of a nonlinear input–output mapping, the multilayer neural network require a smaller number of parameters then the radial basis function (RBF) network for the same degree

of accuracy.

Chapter 6 presents some original work by the extension of the two-stage model to the transient engine mapping application. The modification to the general structure at stage–1 in the hierarchy is done to allow the presence of *forcing function* or *identification signal* at stage–1 in the model. A transient throttle body airflow characterisation is performed using a data from a transient engine test bed. Transient model for a mass airflow is generated from the design data. A biological growth profile is fitted to the data in level-1 with time as a first covariate through a separate Simulink®model developed. The model is validated with a separate set of data with good predictive results.

Chapter 7 describe the application of the transient model-based engine model to the *characterisation problem* (populating look-up tables). The calibration of inferred airflow profiles is presented, and a general method of generating accurate look-up representations of any response surface is discussed. Finally, Chapter 8 presents the conclusions and recommendations for future work.

The flowchart Figure 1.1 gives an outlook for reading this thesis. The motivation, problems and objectives are discussed in Chapter 1. Chapter 2 and 3 gives review and analysis of model based methodology, with discussion of designed of experiments and, modelling and fitting methods. The concept of two-stage regression approach for engine calibration and its mathematics is also discussed. Chapter 4 and 5 are intended to illustrate the two-stage regression methodology with the development and validation of steady state engine model using neural network. Chapter 6 discuss the modification of two-stage regression approach for solving transient calibration. The calibration generation is discussed in Chapter 7, with final conclusion and future work in Chapter 8.

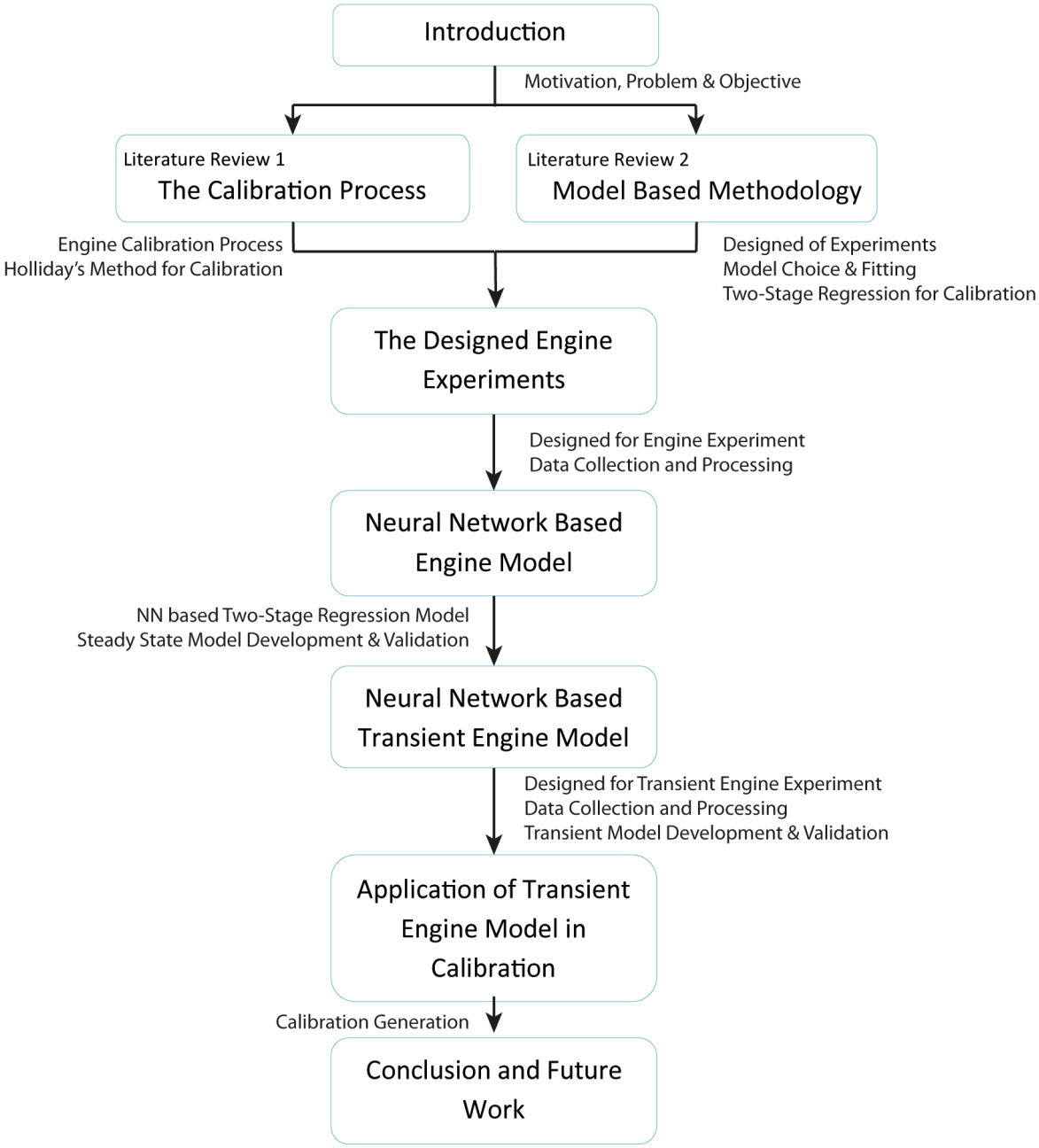


Figure 1.1: Flowchart for reading this thesis

2

Literature Review 1: The Calibration Process

2.1 Introduction

Engine mapping is an empirical modelling of the behaviour of an internal combustion engine (ICE) over a wide range of operating conditions (48; 49; 50; 51), and is a common process in ICE development. Engine data are normally collected in large number of operating points in a dynamometer cell equipped with data acquisition system for search of the optimal set point combination of all independent control variables. These control variables are embedded in the engine management system (EMS), along with various actuators and sensors. All the actuators and sensors in the engine management system are controlled through engine Electronic Control Unit (ECU) to allow the state of the power train to be continually monitored and altered when required. A large number of operating parameters have to be detected with the aid of sensors, and processed using algorithms – i.e. a set of defined mathematical rules. The results obtained take the form of sequence of signals which are used to control actuators. A control program or *strategy* is executed in a continuous loop to determine and set the desired state. Figure 2.1 shows a schematic of a typical engine management system.

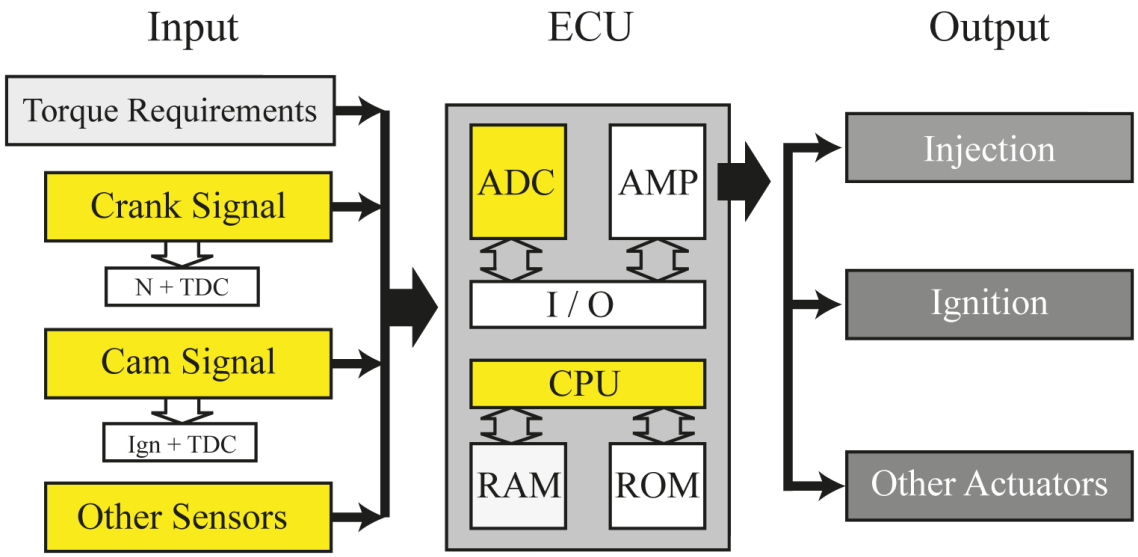


Figure 2.1: Overview Engine Management System (EMS)

In conventional injection systems, for example, the driver directly controls the throttle-valve opening through accelerator pedal. In doing so, he/she defines the amount of fresh air drawn in by the engine. In engine-management systems with electronic accelerator pedal for cylinder-charge (also known as EGAS or ETC/Electronic Throttle Control), the driver inputs a torque requirement through the position of the accelerator pedal, for instance to fulfil the demand to accelerate. Here, the accelerator-pedal sensor measures the pedal's setting, and the ETC subsystem uses the sensor signal to define the correct cylinder air charge corresponding to the driver's torque input, and opens the electronically controlled throttle valve accordingly. These signals are provided by interrogating a brake torque response surface model stored in the ECU in lookup table form. These lookup tables are populated or calibrated using predictions from response surface models developed during the course of engine mapping study.

These lookup tables form the basis of the control strategy, and are two dimensional arrays indexed by the appropriate state and actuation variables.

2.2 The Engine Calibration Process

The actual engine mapping process consists of many different steps ranging from creation of engine map to developing calibrations for specific applications (52). This characterisation takes place traditionally with identification of the engine operating region, in terms of speed and load during the test cycle of interest. The design of mapping test can be specific to a particular application, or general, if whole ranges of applications are required. However, with the increased number of variables, the data required to calibrate the engine increases exponentially. Hence, testing on a full factorial results in very high number of operating points, and conventional methods of calibration and optimization are now entirely impossible to implement (43; 53). As a result, there is a growing realisation that the model based calibration can reduce burden. The model based approach to calibration solves this problem of exponential scaling of testing time with number of control parameters by using design of experiments (53). The method is based on building a statistical model of empirical data to capture the engine behaviour, which help in the reduction of the data collection for building the model. This gives a linear dependence of testing time on the number of control parameters rather than the exponential dependence. These models are further used to generate the optimised calibration tables for the electronic control unit (ECU), for both control and estimator problems. Figure 2.2 shows a calibration process also known as Z-process (54), which consists of the five sub-processes 'Definition of factors and responses', 'Experimental design', 'Measurement on the test bench', 'Modelling', 'Calibration and Optimization' and 'Filling tables of ECU'. The experimental plan is devised through application of advanced DoE methods. Further, statistical modelling uses data collected from the experimental plan to produce accurate response models. Finally, high quality engine calibrations are then developed through optimization of these models and system and calibration verification.

Traditional calibration methods have focused on optimizing a single variable at a time on the engine test bed which often negates the interaction between other

input variables. With higher degrees of freedom this becomes a very time consuming and inefficient process. Engine mapping process is considered heavily a statistical method (48; 49; 52) due to the empirical nature of the data, and regression analysis are commonly used for building empirical models. With the introduction of the model based calibration approach it has made possible to optimize all degrees of freedom simultaneously to enable a complete systems approach. DoE is the major part in statistical models, because of the fact that the effect of interaction between calibration settings and engine performance can be well explain by these models, which is vital to optimal control of the engine. Optimal engine calibrations can be generated from the models for maximum performance, driveability and different constraints.

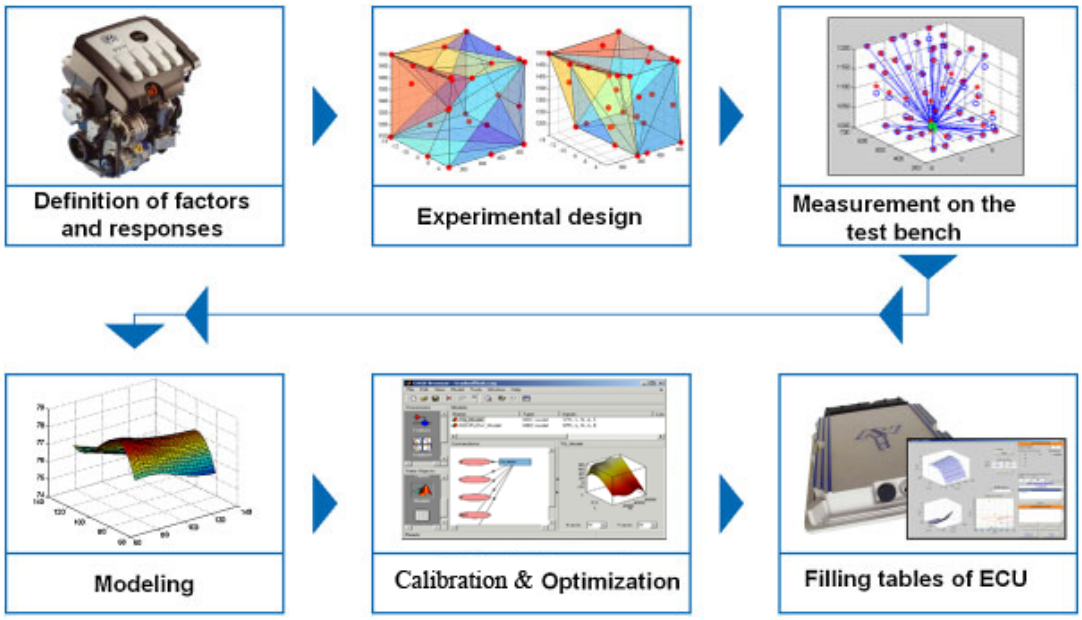


Figure 2.2: Steps of a model-based calibration process

2.2.1 Holliday's Approach for Engine Mapping

The potential of two-stage regression method for engine mapping process in calibration process has been investigated by Holliday (49; 50). This unique method

is based on the fact that the engine data for the mapping process are organized in sweeps, and the empirical modelling benefits more from the two-stage of the data. This two-stage model simplifies the data analysis, such as model selection, validation and diagnostics, and hence provides a framework which is more readily interpreted by engineering analysts. The method provides a very good predictions and a major justification of this approach. The sequence of tasks undertaken in this approach is shown in Figure 2.3

The main aim of experimental design is to select a small subset of possible input configuration at which engine may function. These information content carried by the experimental data at design points is assumed sufficient to illustrate the true behaviour of the response surface of interest. Once the data is collected the information content in that data is fixed. So it is important to design the experiment prior to the data collection. Any model that is built on the insufficient data will yield an adequate prediction, no matter how sophisticated the analysis is.

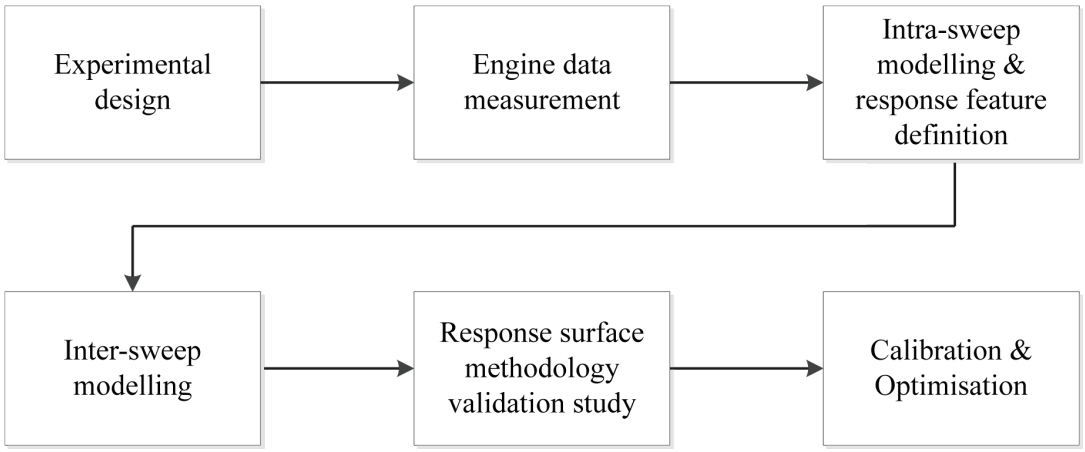


Figure 2.3: Two-stage regression approach to engine mapping

During the engine mapping experiments, the data are normally collected in sweeps. The process involves holding the entire engine operating parameter constant, while sweeping one variable from one extreme to another in discrete steps.

This results in a *cluster* of data in *sweeps*, which comprise the fundamental experimental units.

Holliday (49; 50) first suggested the use of non-linear mixed effect models for the analysis of *repeated measurement* data (55). Crowder and Hand (56) defined the repeated measurements that are made of same characteristic on the same observational unit but on more than one occasion. In engine mapping context, the repeated measurement are the data sweeps. The earlier approaches in engine mapping were based on polynomial regression (52; 57), and this method was not explored.

Modelling is divided into two-stages: *intra-sweep* and *inter-sweep* modelling. Intra-sweep is concerned with the variation *within* a given variable or, the relationship between the response of interest and the swept variable. Each time a sweep is repeated a slightly different profile is obtained, even when the settings of the remaining experimental variables are not varied. A non-linear fit function summarise the relationship for each sweep, while the corresponding regression parameters characterise the shape of the sweep-specific response profile.

The inter-sweep modelling is concerned with the variation *among* the variables or, the relationship in the sweep specific parameter vectors with the remaining engine operating variables. The second stage is multivariate in nature as contrast to the first stage which deals with the relationship of one variable with the response.

The hierarchical modelling techniques has an advantage from an engineering perspective, as the curve fit does not have any intuitive interpretation, rather their characteristic geometric features which is of interest. The second stage of the modelling is conducted in terms of *response features* (56) rather than the sweep-specific regression parameters themselves.

Once the model is developed, it should be tested to check the behaviour in accordance with the physical or empirical laws established for proper function of the system. This validation of model is referred as *internal validation*. In addition,

the model should predict the response characteristics at previously untried input configurations not included in the training data, the *external validation*. If the model is not sufficiently accurate then it may be augmented, as the resultant calibrations can only be as good as the response surface model.

After a sufficient external validation, the model is used for the generation of engine control strategy calibration. This involve the *characterisation*, populating the lookup tables, and *optimisation*, determining the optimal schedule for control problem.

The characterisation problem is straightforward, which involves selecting the table values and the corresponding indexing scheme, or table *break-points*. The objective of the characterisation problem is the selection of break-points and the corresponding table values so that the fit to the relevant response surface projection onto the table domain is optimised. The final lookup table is the representation of the response surface by a control strategy. The strategy should be improved, if the representation is insufficient and differences occur in certain operating regimes.

A multi-objective optimisation problem is formulated in the presence of competing objectives, which have to be trade-off against each other. In this case, one objective is optimally satisfied at the expense of others. However, a multitude of potential trade-off are evaluated in coming to a decision.

There are two valuable benefits from the Holliday's two-stage approach. First, it gives the engineer an opportunity to have a clear understanding of the factor being varied, at each operating condition at a second stage. This leads to easier identification of outliers within the test data. Second, the complexity of the designed experiments can be reduced by one factor, which will lead to reduced model complexity. Performing sweeps will add to the total number of test points, but the above benefits far outweigh this disadvantage. Each spark sweep may be considered as a local model or first stage, and each DoE test point as the global model or second stage'.

2.2.2 Limitation of Holliday's Approach

Rose et al (2) used the two-stage technique in selecting more realistic models and their corresponding experimental design protocols, in contrast with the Holliday method where the model was limited to only small data set collected over a limited range of engine operation. Their method demonstrates the effectiveness of the 2-stage modelling approach when employed to map engine stability and combustion performance parameters. Cary (58) and Guerrier (36) demonstrated the use of two-stage methodology to a realistic calibration problem. They have also used different modelling approaches, such as Hybrid B-Spline and radial basis function (RBF) for the regression analysis. Tindle (59) demonstrated the methods for cold engine emissions optimization.

However, the two-stage model discussed by the researchers mentioned above assumed that the covariate vector in the second stage summarizing individual characteristics is constant across the observations on individual, and which further specifies that the value of the regression parameter of first stage for individual remains fixed for that individual over the course of observation. In some cases, particularly in engine transient phenomena such as engine warm up and fuel dynamic response characteristics, individual specific information may change during the course of observations to exhibit corresponding changes at different time.

2.3 Summary

Engine calibration process is discussed in this chapter, with reference to different approaches. The main aim of all these methods is their performance and predictive qualities. Holliday's method of engine calibration process is based on the development of empirical model; with the realisation of the data structure compose in sweeps. The two-stage regression approach has good fitting and predictive capabilities than the other polynomial models used previously.

However, although two-stage regression methods were analysed for the engine mapping data, but all the related work is limited to the application of the method

to only steady stage condition. However, the steady state engine mapping or calibration optimisation are not suited to the situation in which the prevailing emissions standard is transient standard, which implies that the engine calibration optimisation must be performed using transient or dynamic engine model.

This lead to the research problem for the development of transient engine model for engine calibration, based on two-stage regression approach for the transient engine calibration application.

3

Literature Review 2: Model Based Methodology

Models have always been used in engineering. There are several definitions of what a model is. The concept of model can be defined as:

A model is a simplified representation of a real or imagined system that brings out the essential nature of this system with respect to one or more explicit purposes.

Models are used implicitly in the mindset of the engineer, in terms of construction of physical models/prototypes. Much of today's engineering analysis consists of running complex experiments: applying a vector of design variables (inputs) x and measuring a vector of responses (outputs) y. However, due to increase in the number of variables and the response associated with them, the expense of running an experiment remains non-trivial; a single experiment run involving a few variables can take minutes to hours, if not longer. Moreover, this mode of query-and-response often leads to a trial and error approach to design, whereby a designer may never uncover the functional relationship between x and y, and therefore never identify the 'best' settings for input values.

Statistical techniques are widely used in engineering design to address these con-

cerns. The basic approach is to construct approximate model that are close to real one but more efficient and fast to run, and yield insight into the functional relationship between x and y. If the true nature of an experiment is:

$$y = f(x)$$

then the approximate model is;

$$\hat{y} = g(x) \text{ and so } y = \hat{y} + \epsilon$$

where ϵ represents both the error of approximation and measurement (random) errors. These type of methods are widely used in design and analysis of computer experiments, and are referred as *'model of the model'* or *metamodel* (1; 60). Here, in engine development context, this can be easily fitted in 'Model based methods', which is a combination of first principles, equation based modelling and data-based techniques to develop high fidelity, real-time dynamic model for predicting engine emissions, performance and operating states (43). This approach involves:

- Choosing and experimental design for generating data
- Choosing and fitting the model to the data

Model is seen as a vital element in research and development, as it may be regarded as a solution of a set of equations, including linear, nonlinear, ordinary, and/or partial differential equations, and it is often impossible to obtain an analytic solution for the equations. These models are always crucial in a situation where a relationship is needed between a response y and inputs x's and performing physical experiments are too expensive or time consuming to conduct.

There are several options available for construction of a model, Simpson *et al.* (60) and Wang and Shan (61) highlighted a few of the most frequently used ones. These are shown in Table 3.1. For example, building a neural network involves fitting a network of neurons by means of back-propagation to data which is typically based on Latin hypercube sampling, while Response Surface Methodology

(RSM) usually employs central composite designs, second order polynomials and least squares regression analysis. Also, guidelines and recommendations are given in (60).

Table 3.1: Model Based Methods: Experiment design, model choice and fitting

Experimental Design	Model Choice	Model Fitting	Sample Approximation Techniques
Fractional Factorial	Polynomial (Linear, Quardatic)		
Central Composite	Spline (Linear, Cubic)	Least Square regression	Response Surface Methodology
Box Behnken	Multivariate Adaptive		
Optimal	Regression Spline (MARS)	Best Linear Unbiased Predictor	
Plackett-Burman			Kriging
Hybrid Methods	Kriging	Multipoint approximation	
Sequential Methods	Radial Basis Function (RBF)		
Latin Hypercube	Artificial Neural Network (ANN)	Back Propagation (for ANN)	Neural Network
Orthogonal Array			
Select by Hand	Decision Tree	Entropy (for Inductive Learning)	
Random Selection	Hybrid Models		Inductive Learning

A review of the methods used is given in next sections. Section 3.1 presents the basic concepts related to experimental design and a survey to the design methods used. In Section 3.2 different modelling techniques including regression, neural networks, inductive learning and kriging are discussed. In Section 3.3, two stage models for engine is described.

3.1 The Designed Experiments

Design of Experiment (DoE) is forming a detailed experimental plan in advance for doing the experiment. Experimental designs minimize the number of test points and maximize the amount of information that can be obtained for a given amount of test. DoE is now commonly used in engine development for the reason of increasingly complex and sophisticated modern engines, which demand high calibration and optimization effort.

The use of experimental design is not a new concept. Other industries such as agriculture have been utilising them since the early 1920's, which started with the pioneering work of Sir Ronald A. Fischer. Fisher developed an insight that

led to the three basic principles of experimental design: randomization, replication and blocking. His work introduced a statistical thinking and principles into designing experimental investigations, including factorial design concept, which is the basic building block of subsequent work in the field.

Box and Wilson (62) spark the new era of statistical design with their development of response surface methodology (RSM). Their recognition to the fact that many experiments are fundamentally different than their agricultural counterparts in two ways: the response variable can usually be observed (nearly) immediately and, the experimenter can quickly learn crucial information from a small group of runs that can be used to plan next experiments. These techniques were spread to other research and development work, mainly chemical and process industries.

The third era started with introduction of statistical designs in quality improvement in western industries in late 1970s, and the work of Genichi Taguchi (63) played an important role in that. He suggested highly fractionated factorial designs and other orthogonal arrays along with some novel statistical methods in the field. Although Taguchi concepts and objectives were well founded but there were substantial problems with his experimental strategy and methods of data analysis, and that led to many controversies. The positive outcome of the Taguchi was, that statistical designs spread to other industries including automotive and aerospace, and also it started the fourth era of statistical design. This era has included a renewed interest in statistical design by both researchers and practitioners. There are plenty of standard texts that discuss the statistics and theory behind the concept (12; 64; 65).

However, engineers tend to shy away from the statistical theory such that additional texts that deal with the practicalities of experimental design have emerged (66). The application of experimental design within the automotive industry for engine, emissions and fuel economy optimisation has been around for a little more than a decade. Applications include, optimisation of cold start emissions (67), inlet port design (68), catalyst system optimisation (69) and the optimisation

of variable valve trains for performance and emissions (70; 71) as well as many others.

3.1.1 Basic Concepts in Experimental Design

Design of experiments is an efficient procedure for planning experiments so that the data obtained can be analysed to yield valid and objective conclusions and to derive a response relationships, describing it in term of factors (48; 72). 'Factors' are one or more independent variables, in any experiment that are varied at each test points in order to observe the effect on one or more dependent parameters, known as 'response'. These factors are set to specific values, or 'levels' in these tests which specify the range over which each of them is to be tested. At each test point, the objectives of an experiment and selecting the factors for the study are to be determined. Some basic concepts relating to DoE methodology are discussed as under:

Factor: A factor or input variable is a controllable variable or parameter that is of interest, and is eligible to be analysed in different levels during the experiments. A factor may be quantitative or qualitative. A quantitative factor is one whose values can be measured on a numerical scale and that fall in an interval, e.g., temperature, pressure, etc. A qualitative factor, also known as categorical factor or indicator factor is one whose values are categories such as different operators, different material, etc. All those factors that cannot be controlled (sometimes an experimental error), but has got influence on the experimental data, are known as noise factors.

Experimental domain, level, and level-combination : Experimental domain is the space where factors (input variables) take values. In experiments, experimental domain is also called input variable space. A factor can be tested in experimental domain for some specific values, called as levels of the factor. A level-combination (also called treatment combination) is one of the possible combinations of levels of the factors. A level-combination can be considered as a point in input variable space and called experimental point.

Run, trial: An implementation of a level-combination in the experimental environment is known as run or trial. This can be physical experiment or computer experiment. But the word 'trial' is only meaningful in physical experiments, as in computer experiment several run at same experimental point will result in same data.

Response: Response is a result obtained by each combination of levels of the input factors. The response can be numerical value, qualitative or categorical; and can be a function that is called functional response. Responses are also called as outputs in many computer experiments.

In the next section an overview of the different types of experimental design is presented, along with measures of merit for selection comparing different experimental design.

3.1.2 A Survey of Experimental Design

A proper experimental design is essential for an effective utilisation of the system, and hence, provides a representative data for the model generation. An experimental design is represented by a matrix with rows denote experimental runs and the columns denote particular factor settings.

3.1.2.1 Classical Designs

Much of the work that utilises experimental design involves what is now known as *classical* design of experiments. This focus on planning experiments so that the random error in physical experiments has minimum influence on the approval or disapproval of a hypothesis. Typically, these consist of 2 or 3 level experiments that exhibit 1^{st} or 2^{nd} order polynomial responses. These widely used classical experimental design include factorial or fractional factorial, central composite design (CCD), alphabetical optimal, and Box- Behnken designs as well as others (73). These classic methods tend to spread the sample points around boundaries of the design space and leave a few at the centre of the design space. Much effort

was placed into reducing the number of tests required, where for large numbers of variables the higher order interactions were considered insignificant.

Factorial Design Factorial design is a set of level combinations necessary to study effect of the factors on a response. A factorial design is called symmetric if all factors have the same number of levels; otherwise, it is called asymmetric.

One of the important type of these factorial designs is that of k factors, having only two levels. A complete replicate of the design requires 2^k observations and it is called as a 2^k factorial design. When n replicates of each treatment are necessary there are $n2^k$ observations (65). Factorial design is particularly useful in the early stages of experimental work, when it is likely to investigate many factors.

Full factorial design A design where all level combinations of the factors appear equally often is called a full factorial design or a full design. The number of runs of a full factorial design increases exponentially with the number of factors. Therefore, implementation of a subset of all level-combinations that have a good representation of the complete combinations is considered.

Fractional factorial design A fraction of a full factorial design (FFD) is a subset of all level-combinations of the factors (64). Fractional factorial designs are used when experiments are costly, and many factors are required. A fractional factorial design is a fraction of a full factorial design; the most common are $2^{(k-p)}$ designs, in which the fraction is $1/2^{(p)}$. A half fraction of the 2^3 full factorial design is shown in Figure 3.1(b).

Reduction in the number of design points in a fractional factorial design is always with some compromise in desired response limitation. The 2^3 full factorial design shown in Figure 3.1(a) allows estimation of all main effects (x_1, x_2, x_3) all two factor interactions (x_1x_2, x_1x_3 and x_2x_3), as well as the three factor interaction ($x_1x_2x_3$). For the 2^{3-1} fractional factorial indicated by the solid dots in Figure 3.1(b), the main effects are aliased (or biased) with the two factor interactions.

Aliased effects cannot be estimated independently unless they are known (or assumed) not to exist.

A carefully FFD selected combination known as the orthogonal array is recommended in the literature and has been widely used in practice.

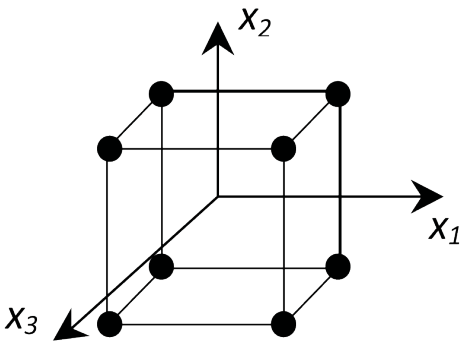
Orthogonal Arrays The experiment designs used by Taguchi, orthogonal arrays, are usually simply fractional factorial designs in two or three levels ($2^{(k-p)}$ and $3^{(k-p)}$ designs. These arrays are constructed to reduce the number of design points necessary; two-level L_4, L_{12} and L_{16} arrays, for example, allow $3, 11$ and 15 factors/effects to be evaluated with $4, 12$ and 16 design points, respectively. Often these designs are identical to Plackett-Burman designs (74).

Central Composite and Box-Behnken Designs Even with the optimum designs, higher numbers of levels or variables witnessed a sharp increase in the number of points required (36). Also, to fit higher order responses, higher degrees of freedom would be required. The most common second order designs, configured to reduce the number of design points, are central composite and Box-Behnken designs.

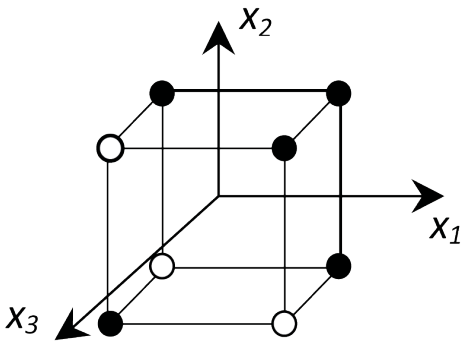
Table 3.2: Classical Central Composite Design

Variable	Number of Test Points			Total
	Base	Star	Centre	
2	4	4	5	13
3	8	6	6	20
4	16	8	7	31
5	16	10	6	32
6	32	12	9	51
7	64	14	14	92

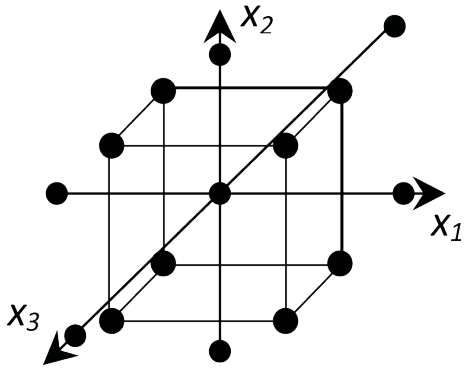
A central composite rotatable design could be used where additional star and centre points would augment the base test matrix. Figure 3.1(c) shows a schematic of a Central Composite Rotatable Design (CCRD) that comprises a three level, three variable designs augmented with 6 star points and a centre point. Increasing the number of repeated centre points would reduce the standard error of predicted response near the centre. Table 3.2 shows the number of points required for a CCRD designs based upon a Hadamard base design.



(a) 2^3 full factorial



(b) 2^{3-1} fractional factorial



(c) Composite design

Figure 3.1: Basic three-factor designs

A Central Composite Design (CCD) is a two level ($2^{(k-p)}$ or 2^k) factorial design, augmented by n_0 center points and two 'star' points positioned at $\pm\alpha$ for each factor. This design, shown for three factors in Figure 3.1(c) and 3.2 consists of $2^{(k-p)} + 2^k + n_0$ total design points to estimate $2k + k(k-1)/2 + 1$ coefficients. For three factors, setting $\alpha = 1$ locates the star points on the centres of the faces of the cube, giving a face-centred central composite (CCF) design; note that for values of a other than 1, each factor is evaluated at five levels.

Box-Behnken designs are used when the smallest number of factor levels in an experimental design is required. These are formed by combining 2^k factorials with incomplete block designs. They do not contain points at the vertices of the hypercube defined by the upper and lower limits for each factor. This is desirable if these extreme points are expensive or impossible to test. More information about CCD and Box-Behnken designs can be found in Montgomery (65).

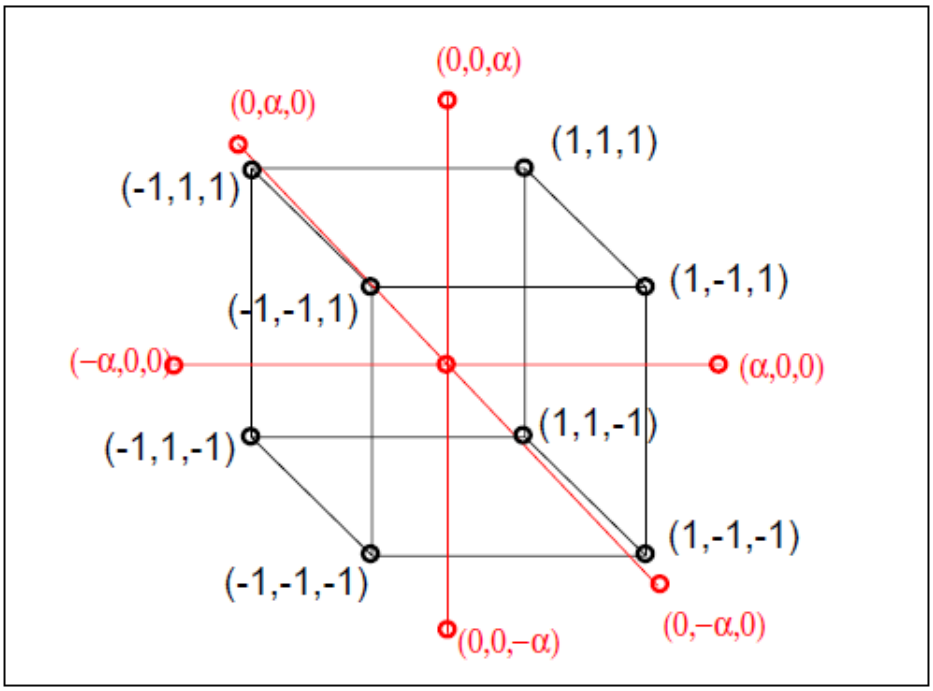


Figure 3.2: CCRD design space

The main drawbacks when using classical designs are the fact that they are

not very flexible, each point must be visited, and failure to accomplish these results in the design being compromised. Therefore the boundaries must be identified correctly and the experimental space potentially reduced (to ensure the recommended equidistant points from the centre) to ensure valid data points. Whilst the main testing may exhibit relatively few points, large amounts of pretesting (screening) would be required in order to identify the experimental space (constraints region) accurately. For engine testing, the constraints region is not symmetrical and difficulties in setting up each factor may witness large amounts of screening.

Second orders polynomial equations are commonly used in construction of experimental model, and are applied in a wide range of application in automotive engineering, from design and development to calibration and production. However, this can lead to restricted variable ranges to improve the fit. With increasing levels of technology, the number of operating points grows exponentially, and the limitation of using these second orders polynomial based models increase. These thus reduce the classical designs only to linear modelling methods. However, these points should not preclude the use of classical DoE's, providing some consideration goes into selecting the number of variables and their ranges, many successful designs have been implemented within the field of IC engine optimisation. If DoE is to be applied to multi-dimensional non-linear systems, more advance (sophisticated) techniques need to be considered (75). Many different techniques are available that offers non-linear capability, statistical information and computational requirements.

3.1.2.2 Space Filling Designs

The classical design methods tend to spread the sample points around boundaries of the design space and leave a few at the centre of the design space. This would not cover the experiments that involve mostly systematic error rather than random error as in physical experiments. Sacks et al. (76) stated that in the presence of systematic rather than random error, a good experimental design tends to fill the design space rather than to concentrate on the boundary. They also

stated that classic designs, e.g., CCD and D-optimality designs can be inefficient or even inappropriate for deterministic analysis. Jin *et al* (77) confirmed that a consensus among researchers was that experimental designs for deterministic analyses should be space filling.

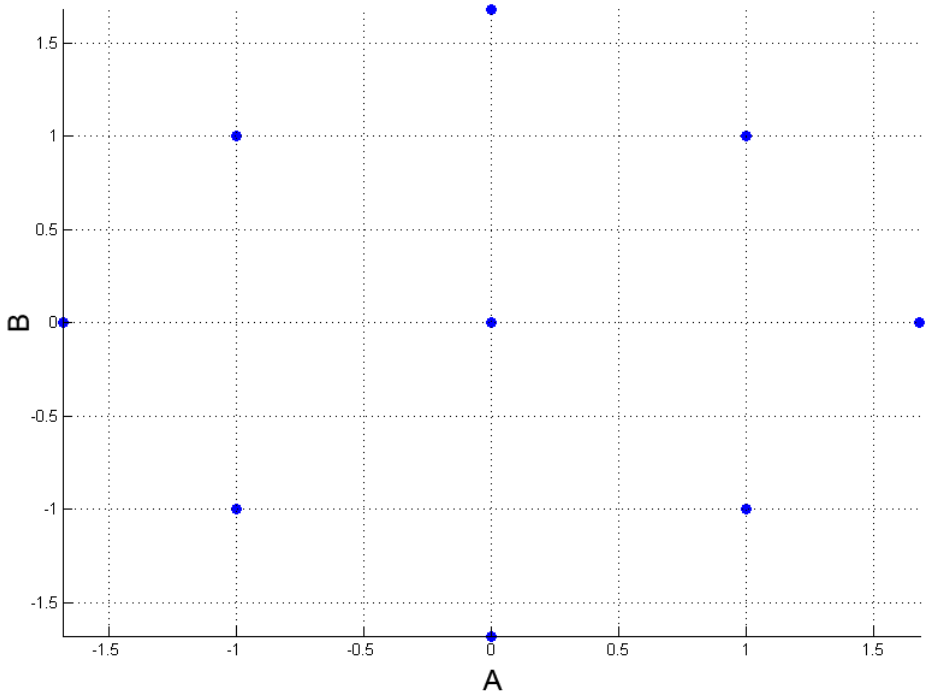


Figure 3.3: CCRD data point projection

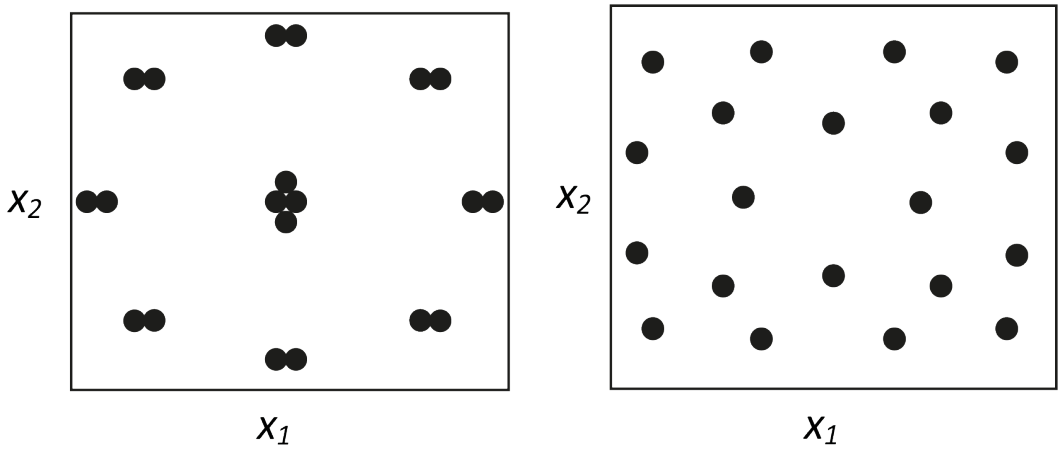
If the data points from the CCRD design are projected onto a two-dimensional plane, as shown in Figure 3.3, it can be seen that the factors (A & B) only exhibit 5 discrete values. If the factor range were large, it would be possible to miss some significant interaction. For the same number of test points, the use of advanced DoE/modelling techniques would allow 20 levels for each of the factors, thereby ensuring that any interaction could be identified.

Koehler and Owen (78) described several Bayesian and frequentest "space filling" designs, including maximum entropy design, mean squared-error designs, minimax and maximin designs, Latin hypercube designs, orthogonal arrays, and

scrambled nets. However, four types space filling sampling methods are relatively more often used in the literature. These are orthogonal arrays, various Latin hypercube designs, Hammersley sequences, and uniform designs (60; 61). The Latin hypercube design is only uniform in one-dimensional (1D) projection while the other methods tend to be more uniform in the entire space. Also a suitable sample size depends on the complexity of the function to be approximated, with more sample points offer more information of the function, however, at a higher expense. After reaching a certain sample size, increasing the number of sample points does not contribute much to the approximation accuracy of the low-order functions. Moreover, when certain optimality criteria are used to generate samples, these optimality criteria such as maximum entropy are concerned with the sample distribution and are independent to the function. While the approximation accuracy depends on whether sample points capture all the features of the function itself. Therefore, those optimality criteria are not perfectly consistent with the goal of improving approximation, due to which the additional computational cost of searching for the optimal sample is often not well justified.

In the classical design and analysis of physical experiments, random variation is accounted for by spreading the sample points out in the design space, and by taking multiple data points (replicates) 3.4(a). Sacks *et al.* (76) state that the 'classical' notions of experimental blocking, replication and randomization are irrelevant for some experiments; thus, sample points should be chosen to fill the design space. They suggest minimising the Integrated Mean Squared Error (IMSE) over the design region by using IMSE-optimal designs; the 'space filling' design illustrated in Figure 3.4(b) is and IMSE design. Simpson *et al* recommend the use of space filling designs in the early stages of design when the form of the model cannot be pre-specified.

The significant advantages of using the advanced designs are the fact that it eliminates the complexities of the classical design since the high level interpretation is to fill a space with data points and to utilise the flexibility of the advanced models to fit a response to the data collected. Virtually any complex surface can be realised providing enough data points are collected.



(a) Classical **(b)** Space Filling

Figure 3.4: Comparison between 'Classical' and 'Space filling' designs

This means that the identification of the constraints region is potentially less critical than that associated with classical DOE's. Conversely, the use of the advanced design techniques witnesses a reverse trend in the number of data points required. Increasing the number of test points increments both time and cost to the engine mapping process. This can be alleviated somewhat by the use of automated mapping. Another potential downfall is the fact that it is very easy to over-fit models in order to achieve satisfactory response model fit statistics where any noise would be incorporated into the response surface. However, the flexibility and advantages of utilising advanced techniques far outweighed their disadvantages such that they were incorporated into the calibration methodologies.

Latin Hypercube Sampling Latin Hypercube Sampling (LHS) was introduced by McKey, Becham and Conover (79). LHS is an extension of stratified sampling which ensures that each of the input variable has all portion of its range represented (76). The work of McKey, Becham and Conover (79) show that the

LHS has a smaller variance of the sample mean than the simply random sampling. In stratified sampling the range space R of x can be arbitrarily partitioned to form n strata of equal marginal probability $1/n$, the sample once from each stratum. In Latin hypercube sampling the partitions are constructed in a specific manner using partitions of the ranges of each component of x, where the components of x are independent.

In fact, the LHS can be defined in terms of the Latin hypercube design (LHD), as an n runs into s input variables ($n \times s$) matrix, denoted by $LHD(n, s)$, in which each column is a random permutation of $\{1, 2, ..., n\}$.

An LHS can be generated by an algorithm which first independently take s permutations $\pi_j(1), ..., \pi_j(n)$ of the integers $1, ..., n$ for $j = 1, ..., s$. And then take ns uniform variates $U_k^j \sim U(0, 1), k = 1, ..., n,\ j = 1, ..., s$, which are mutually independent. Let $x_k = (x_k^1, ..., x_k^s)$, where

$$x_k^j = \frac{\pi_j(k) - U_k^j}{n}, \quad k = 1, ..., n,\ j = 1, ..., s \tag{3.1}$$

Then $D_n = \{x_1, ..., x_n\}$ is a LHS and is denoted by $LHS(n, s)$. In the context of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. A Latin hypercube is the generalisation of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it.

For example, consider a case where $n = 4$ and $s = 2$. In the first step two permutations of $\{1, 2, 3, 4\}$ as $\{2, 1, 4, 3\}$ and $\{3, 2, 1, 4\}$ is generated to form an $LHD(4, 2)$.

$$\begin{bmatrix} 2 & 3 \\ 1 & 2 \\ 4 & 1 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 0.3724 & 0.9516 \\ 0.1981 & 0.9203 \\ 0.4897 & 0.0527 \\ 0.3395 & 0.7378 \end{bmatrix}$$

Now the LHS is given by

$$\frac{1}{4}\left[\begin{bmatrix}2 & 3\\ 1 & 2\\ 4 & 1\\ 3 & 4\end{bmatrix} - \begin{bmatrix}0.3724 & 0.9516\\ 0.1981 & 0.9203\\ 0.4897 & 0.0527\\ 0.3395 & 0.7378\end{bmatrix}\right] = \begin{bmatrix}0.4069 & 0.5121\\ 0.2005 & 0.2699\\ 0.8776 & 0.2368\\ 0.6651 & 0.8155\end{bmatrix}$$

Figure 3.5 gives plot of the design in the top left hand pane, where four points are assigned in a grid of $16 = 4^2$ cells, satisfying that each row and column has one and only one point, and each point is uniformly distributed in the corresponding cell. Also shown in figure are three other possible Latin-hypercube designs.

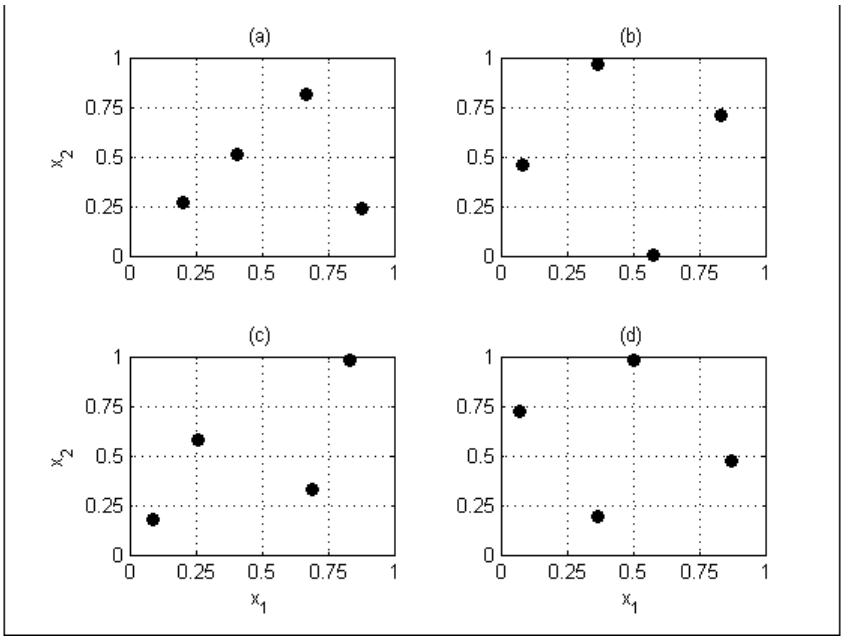


Figure 3.5: Four Latin-hypercube design with four runs

The LHS has many advantages (1), such as:

- Computationally cheap to generate;

- Can deal with a large number of runs and input variables;
- Its sample mean has a smaller variance than the sample mean of a simply random sample.

Many authors tried to improve LHS, i.e. by reducing the variance of the sample mean. One such approach is called *orthogonal array-based Latin hypercube design* (80). The orthogonal sampling adds the requirement that the entire sample space must be sampled evenly. Although more efficient, orthogonal sampling strategy is more difficult to implement since all random samples must be generated simultaneously. In two dimensions the difference between random sampling, Latin Hypercube sampling and orthogonal sampling can be explained as follows:

- In random sampling new sample points are generated without taking into account the previously generated sample points. It is thus not necessarily to know beforehand the quantity of sample points needed.
- In Latin hypercube sampling the sample points quantity to be used is defined; and the location for each sample point in their respective row and column.
- In orthogonal sampling, the sample space is divided into equally probable subspaces. All sample points are then chosen simultaneously making sure that the total ensemble of sample points is a Latin Hypercube sample and that each subspace is sampled with the same density.

Thus, orthogonal sampling ensures that the ensemble of random numbers is a very good representative of the real variability, LHS ensures that the ensemble of random numbers is representative of the real variability whereas traditional random sampling is just an ensemble of random numbers without any guarantees.

An alternative idea for improving the performance of $LHDs$ is to adopt some optimality criterion for construction of LHS. One of such criteria is maximin or minimax distance (81), which

- A minimax design - minimising the maximum distance between the points
- A maximin design - maximising the minimum distance between the points

For a given number of runs and number of input variables, (n, s), the resulting *design space*, denoted by $\mathcal{D}$, can be set of $U(n, n^s)$ or some subset. Let $d(u, v)$ be a distance defined on $T \times T$ satisfying $d(u, v) \geq 0,\ d(u, v) = d(v, u)$, and $d(u, v) \leq d(u, w) + d(w, v),\ \forall u, v, w \in T$. Consider a design $D = \{x_1, ..., x_n\}$ on T.

A minimax design D^* minimises the maximum distance between any $x \in T$ and $D, d(x, D) = max\{d(x, x_1, ..., d(x, x_n)\}$,, i.e.,

$$min_D max_{x \in T} d(x, D) = max_{x \in T} d(x, D^*) \tag{3.2}$$

A maximin design D_* maximises the minimum inter-site distance $min_{u,v \in D} d(u, v)$, i.e.,

$$max_D min_{u,v \in D} d(u, v) = min_{u,v \in D_*} d(u, v) \tag{3.3}$$

This criteria measure how uniformly the experimental points are scattered through the design, and ensure that no point in the domain is too far from the design point. Thus, making it possible for reasonable predictions anywhere in the domain.

3.2 Model Choice and Fitting

In the previous section various types of designs of experiments were introduced. Once the data have been collected from an experiment, the next step is to choose an approximating model and fitting method which describes empirical relationships between the inputs and outputs. The outputs of experiments are deterministic (i.e., no random errors); therefore, the relationship between the input variables and the output variable by the model is described as:

$$\text{output variable} = f(\text{input variables}) \tag{3.4}$$

where f is an unspecified smooth function to be approximated. Many alternative models and methods exist, but here only review of relevant models to the current study is provided.

3.2.1 Polynomial Models

Polynomial Models is most popular in many modelling context including computer experiments. The models employ a polynomial basis where $r_1, \ldots, r_s$ are non negative integers.

$$\begin{aligned} &B_0(x) = 1, B_1(x) = x_1, \ldots, B_s(x) = x_s, \\ &B_{s+1}(x) = x_1^2, \ldots, B_{2s}(x) = x_s^2, \\ &B_{2s+1}(x) = x_1 x_2, \ldots, B_{s(s+3)/2}(x) = x_{s-1} x_s, \end{aligned} \tag{3.5}$$

The number of polynomial basis functions dramatically increases with the number of input variables and the degree of polynomial. Low-order polynomials such as the second-order polynomial model, also known as response surfaces (82; 83), are the most popular polynomial models for experimental modelling. A second-order polynomial model can be expressed as

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{s} \beta_{ii} x_i^2 + \sum_i \sum_j \beta_{ij} x_i x_j \tag{3.6}$$

These models are usually used to find the overall trend of a true model. When the domain is large and true model is more complicated, e.g., there are many local minimums/maximums, so high degree polynomials are needed to approximate the true model. It is important to use linear or second-order polynomial models to narrow the design variables to the most critical ones, when applying these models with a problems having large dimension. In optimization, the smoothing capability of polynomial regression allows quick convergence of noisy functions. In spite of the advantages, there is always a drawback when applying polynomial regression to model highly nonlinear behaviours. Higher-order polynomials can

be used; however, instabilities may arise, or it may be too difficult to take sufficient sample data to estimate all of the coefficients in the polynomial equation, particularly in large dimensions.

Due to the structure of the polynomial basis, as the number of variables and the order of polynomials increase, the number of possible terms in the polynomial basis grows rapidly. Hence, the number of possible candidate polynomial interpolaters grows dramatically. As a result, the required number of data samples also increases dramatically, which can be prohibitive for computationally expensive simulation models. Therefore, the model is usually limit to only linear or up to lower-order models or models with fixed terms. In practice, once a polynomial interpolator is selected, the second stage consists of reducing the number of terms in the model following a selection procedure, such as a stepwise selection based on Cp, AIC, BIC, or o-criterion. The selected model usually has better prediction power, although it may not exactly interpolate the observed data.

3.2.2 Kriging method (KG)

The Kriging method was proposed by a South African geologist, D.G. Krige on analysing mining data. The Gaussian Kriging method was based on his method proposed by Matheron in 1963 for modelling spatial data in geo-statistics.

Suppose that x_i, with$i = 1, \ldots, n$ are design points over an s-dimensional experimental domain T, and $y_i = y(x_i)$ is the associated output to x_i. The Gaussian Kriging model postulates a combination of a known fixed function $f_i(x)$ and departures of the form:

$$\hat{y} = \sum_{i=1}^{s} \beta_i f_i(x) + Z(x) \tag{3.7}$$

where $Z(x)$ is assumed to be a realisation of a stochastic process with mean zero and spatial correlation function given by:

$$Cov(z(x_i), z(xj)) = \sigma^2 R(x_i, x_j) \tag{3.8}$$

where σ^2 is the process variance, and R is the correlation. A variety of correlation functions can be chosen (84). However, the Gaussian correlation function proposed by Sacks *et al.* (76) is the most frequently used.

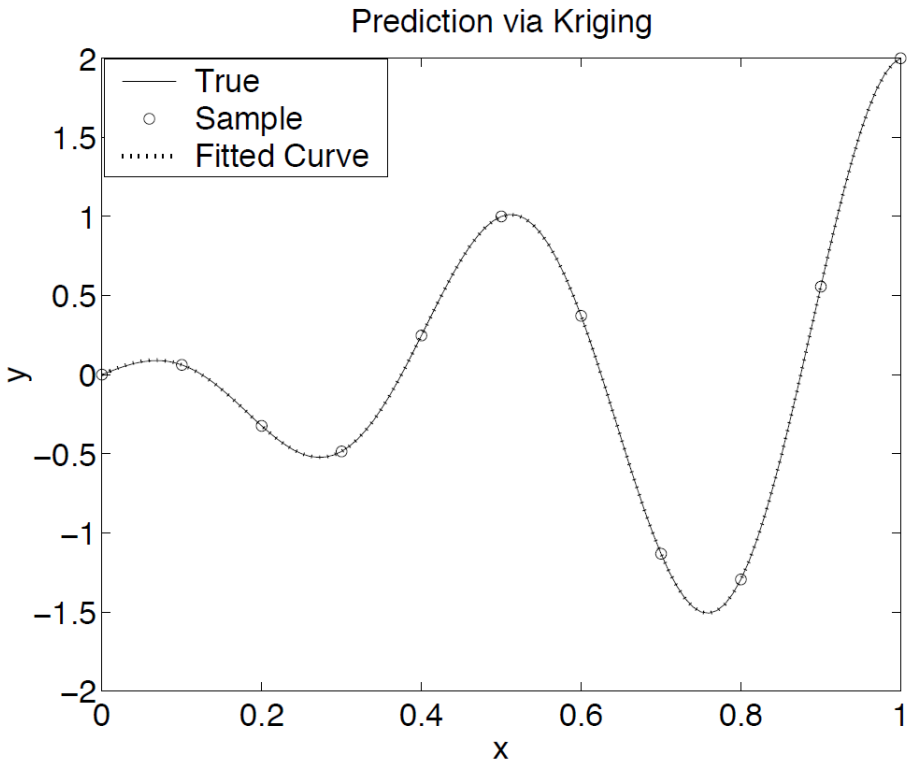


Figure 3.6: Plot of Gaussian Kriging model fit

Figure 3.6 show fit of the data supported by Gaussian Kriging model, that has good prediction by interpolating exactly the observed sample. The predicted curve and the true curve are identical, which compared with the polynomial regression model and regression spline is better.

In addition to being extremely flexible due to the wide range of the correlation functions, the kriging method has to determine the important factors, and the same data can be used for screening and building the predictor model (85). The major disadvantage of the kriging process is that model construction can be very

time-consuming. Determining the maximum likelihood estimates of the θ parameters used to fit the model is a k-dimensional optimization problem, which can require significant computational time if the sample data set is large. Moreover, the correlation matrix can become singular if multiple sample points are spaced close to one another or if the sample points are generated from particular designs. Kriging assumes that the closer the inputs are, the more positively correlated the outputs are. For this reason the fitting problems have been observed with some full factorial designs and central composite designs using kriging models (86). Furthermore, the Gaussian Kriging approach admits a Bayesian interpretation.

Bayesian interpolation was proposed by Currin, Mitchell, Morris and Ylvisker (87). Bayesian interpolation can be beneficial in that it easily incorporates auxiliary information in some situations. Morris (87) demonstrated how to use Bayesian Kriging method to create computer models that can provide both the response and its first partial derivatives.

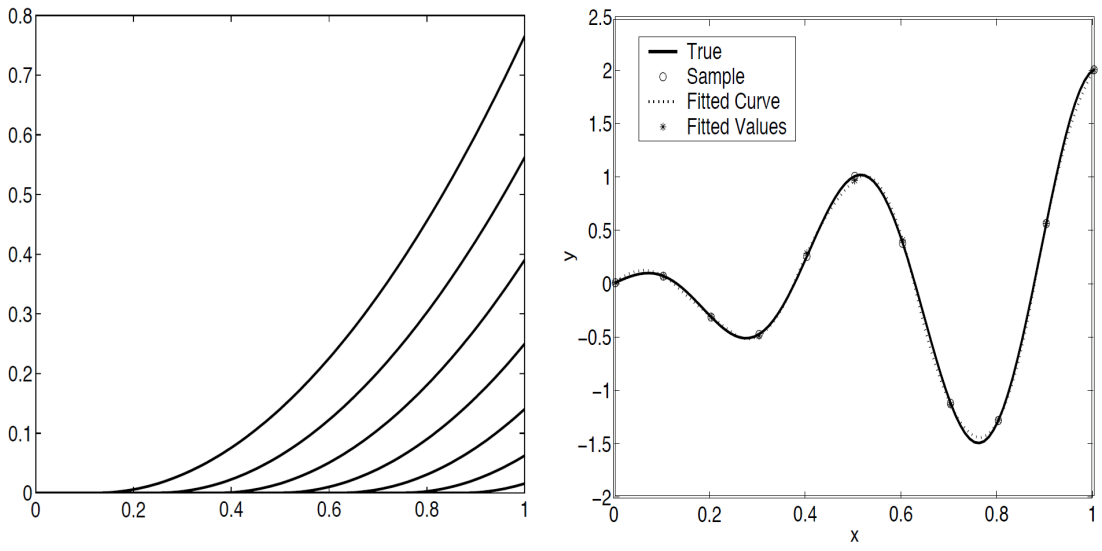
3.2.3 Spline Method

Splines are curves, which are usually required to be continuous and smooth. Splines are usually defined as piecewise polynomials of degree n with function values and first $n-1$ derivatives that agree at the points where they join. The abscissa values of the join points are called *knots*. The term "spline" is also used for polynomials (splines with no knots) and piecewise polynomials with more than one discontinuous derivative. Splines with no knots are generally smoother than splines with knots, which are generally smoother than splines with multiple discontinuous derivatives. Splines with few knots are generally smoother than splines with many knots; however, increasing the number of knots usually increases the fit of the spline function to the data. Splines are frequently used in nonparametric regression in the statistical literature. The spline method mainly includes smoothing splines (88), regression splines (89), and penalized splines (90; 91). The splines can be used to approximate univariate functions as easily as the polynomial regression. The mechanism of constructing a multi-dimensional

spline basis is also similar to that of polynomial regression.

$$s(x) = \beta_0 + \beta_1 x^2 + \beta_2 x^2 + \ldots + \beta_p x^p + \sum_{k=1}^{k} \beta_{p+k}(x - \kappa_k)_+^p \tag{3.9}$$

where $p \geq 1$ is an integer. The $s(x)$ in Equation 3.9 is a p^{th} degree polynomial on each interval between two consecutive knots and has $(p-1)$ continuous derivatives everywhere. Friedman (92), extended one dimensional regression spline fitting to multidimensional model fitting.



(a) Quadratic spline basis

(b) Plot of spline fit

Figure 3.7: Fit of spline method (1)

The fit supported by spline model is depicted in Figure 3.7, from which the linear spline has a slightly better fit to data than the best polynomial fit. A better spline fit may further be pursue by using a higher order spline and/or adding more knots.

Multivariate Adaptive Regression Splines is a flexible regression modelling method based on recursive partitioning and spline fitting for high dimensional data. Regression trees are closely related to MARS. Instead of piecewise-linear approxima-

tion, regression trees form a piecewise constant-approximation. A MARS model can be written as:

$$g(x) = \sum_{i=1}^{n} a_m B_m(x) \tag{3.10}$$

where a_m is the coefficient of the expansion, and B_m, the basis functions, can be represented as

$$Bm(x) = \Pi_{k=1}^{K_m} [S_{k,m}(x_{v(k,m)} - t_{k,m})]_+^q \tag{3.11}$$

where K_m is the number of factors (interaction order) in the m^{th} basis function, $S_{k,m} = \pm 1, x_{v(k,m)}$ is the v-th variable, $1 \leq v(k,m) \leq n$, and $t_{k,m}$ is a knot location on each of the corresponding variables. The subscript "+" means the function is a truncated power function

MARS is used in engineering design applications just recently. Sudjianto *et al.* (93) use MARS to emulate a conceptually intensive complex automotive shock tower model in fatigue life durability analysis. The major advantages of using the MARS procedure appears to be accuracy and major reduction in computational cost associated with constructing the metamodel compared to the kriging method.

3.2.4 Inductive Learning

Inductive learning is one of five main paradigms of machine learning that also include neural networks, case-based learning, genetic algorithms and analytic learning. Of these five, inductive learning is the most akin to regression and meta-modelling. An inductive learning system induces rules from examples; the fundamental modelling constructs are condition-action rules which partition the data into discrete categories and can be combined into decision trees for ease of interpretation (see Figure 3.8).

Training data are required in the form $(x_1, y_1)(x_2, y_2)..., (x_n, y_n)$ where x_i is a vector of attribute values (e.g., processing parameters and environmental conditions), and each y_i is a corresponding observed output value. Although attributes and outputs can be real-valued, the method is better suited to discrete-valued data; real values must often be transformed into discrete representations (94). Once the data has been collected, training algorithms build a decision tree by selecting the 'best' divisive attribute and then recursively calling the resulting data subsets. Although trees can be built by selecting attributes randomly, it is more efficient to select attributes that minimize the amount of information needed for category membership.

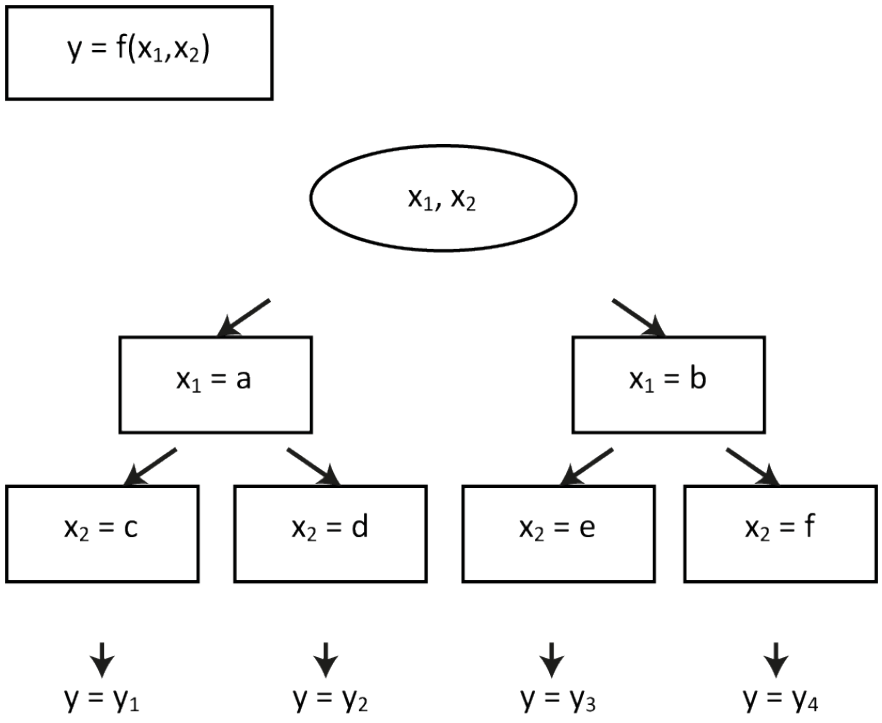


Figure 3.8: A inductive learning decision tree

Many of the applications of inductive learning have been in process control and diagnostic systems, and inductive learning approaches can be used to automate the knowledge-acquisition process of building expert systems. Furthermore, although decision trees appear best suited for applications with discrete input and

output values, there are also applications with continuous variables that have met with greater success than standard statistical analysis.

3.2.5 Neural Network (NN) Models

Artificial neural networks, commonly referred as *'Neural Network (NN)'*, has been motivated by the recognition that the human brain computes in an entirely different way from the conventional digital computer. The brain works in highly complex, nonlinear and parallel computing manner. The basic building blocks of the biological neural system are nerve cells, or *neurons*. These neurons are massively interconnected, with signals transmitting from one neuron to another to its cell body. These neurons can either inhibit or excite a signal. The neural system has the capability to organize these neurons so as to perform certain computations many times faster than the fastest computer in existence today.

As analogue to human neural system, the artificial neuron (AN) receives signals from the environment or other ANs, gathers these signal, and when excited, transmits a signal to all connected ANs. Input signals are inhibited or excited through negative or positive numerical weights associated with each connection to the AN. This excitation is control via a function, referred to as activation function. The neuron collects all incoming signals, and computes a net input signal as a function of the respective weights. The net signal serves as input to the activation function which calculates the output signal of the neuron.

Neural Network has the ability to learn complex, non-linear and multidimensional relationships between multiple input and output variables, resistance to noisy or missing data, with good generalization capability. For example, a neural network can be trained with sufficient data to efficiently replace the engine response functions by predicting response values associated with particular values of engine factors. A more detailed description of the neural network theory and its implementation consideration can be found in (95), but here a brief overview is provided.

Neural Network modelling approach is relatively different from other conventional approaches. NN is purely a data driven technique and learn directly from the data provided, and hence eliminating the use of any mathematical equations or any work with the variables. It is extremely useful in the situation where the system is too complex to find and describe, and when it is too expensive to model the system conventionally (96).

Neural Network consists of several layers of neurons. The first layer is an 'input' layer which in actual do not perform any computation but only transfer its value to the next layer, a 'hidden' layer. The layer that produces output of the network is called an 'output' layer. The input and output layers may be separated by any number of hidden layers, including zero. These layers are arranged by neurons, and each layer may contain any number of these neurons. Neurons only connect the adjacent layers, and signals are propagated via these through each of the hidden layers to the signal layer. The weight and bias associated are adjusted during training, to give a best fit of outputs to the target values in correspondence with the input data.

For a given input vector, it generates the output vector by a forward pass. The data are fed to the network at the input layer, and propagated with weights and activation functions to the output layer to provide the response. After presenting the sets of inputs and associated outputs, the network is able to 'learn' the relationships between them by changing the weights of its connections.

Then, the mean squared error (MSE), the difference between the network output vector and the known target vector, is computed and back-propagated through the ANN to modify the weights for the entire network, a process referred to as training. Learning can be of three types-supervised, unsupervised or reinforcement. One of the most popular methods practised for supervised training of neural networks is the back-propagation training algorithm. The neural networks that are trained by this method are called multilayer feed-forward networks.

The neuron model and the architecture of a neural network describe how a network transforms its input into an output. This mapping of inputs to outputs can be viewed as a non-parametric regression computation. Training in neural networks is synonymous with model building and parameter estimation. There are two popular types of neural network models for performing regression task know as *Multi-layer perceptron (MLP)* and *radial basis function (RBF)* networks.

3.2.5.1 Multi-Layer Perceptron Networks

A single neuron or *perceptron* that consists of inputs, weights and output performs a series of linear and non-linear mapping. The set of n inputs $x_i (i = 1, .., n)$ is processed though the following weight sum,

$$net_i = \sum_{i=1}^{n} w_i x_i + w_0, \tag{3.12}$$

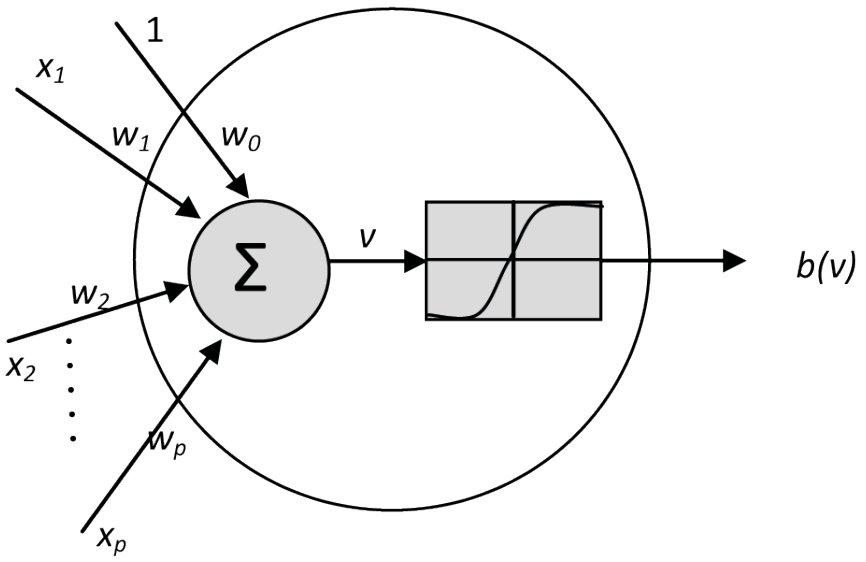


Figure 3.9: A neuron model with x_i as inputs, w_i as weights or parameters, and $b(v)$ as the activation function

and the output of the i^{th} neuron is obtained by processing the weight sum of inputs (Equation 3.12) with a transfer function or activation function, here a logistic-type of function. Figure 3.9 shows a neuron model with x_i as inputs, w_i as weights or parameters, and $b(v)$ as the activation function.

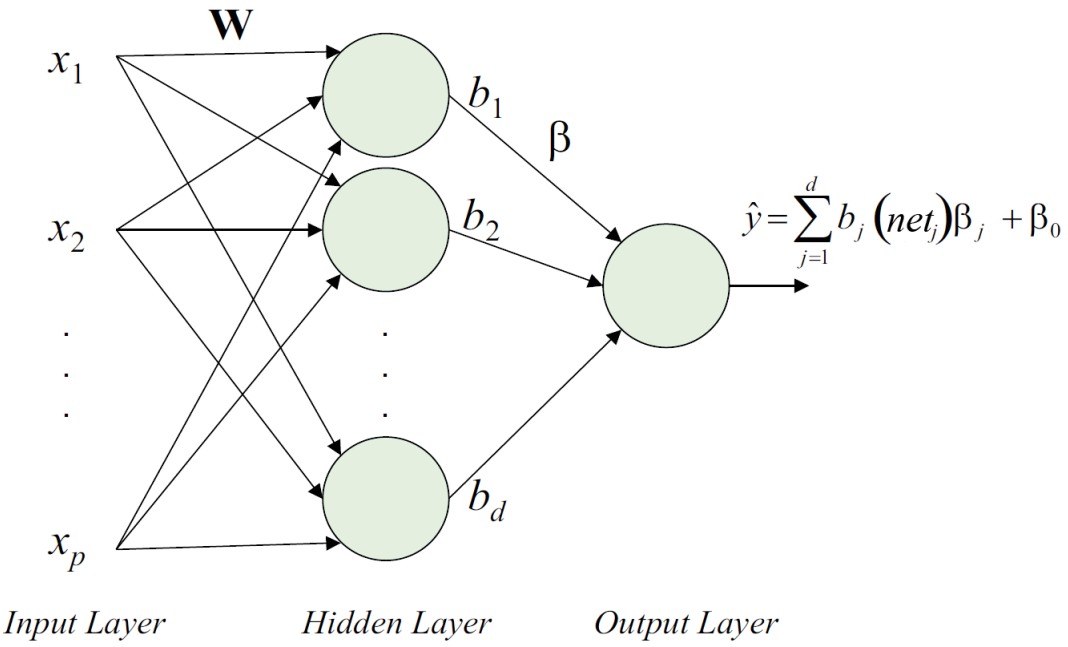


Figure 3.10: A three layer multilayer perceptron (1)

A *multi-layer perceptron*(MLP) network (shown in Figure 3.10, consists of input, hidden, and output layers with nonlinear and linear activation functions in the hidden and output layer, respectively, approximates inputs and outputs as

$$\hat{y} = \sum_{j=1}^{d} \beta_j b_j(net_j) + \beta_0, \tag{3.13}$$

where d is pre-specified integer, β_j is the weight connection between the output and the j^{th} component in the hidden layer, and $b_j(net_j)$ is the output of the j^{th}

unit in the hidden layer,

$$b(net_i) = \frac{1}{1 + e^{-\lambda net_j}} \text{or} b_j(net_j) = tanh(\lambda net_j), \tag{3.14}$$

and

$$net_j = \sum_{i=1}^{s} w_{ji} x_i + w_{j0}, \tag{3.15}$$

where w_{ji} is the weight connection between the j^{th} component in the hidden layer and the i^{th} component of the input.

Multilayer perceptrons have been successfully applied to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the *error back-propagation algorithm* or *back-propagation*. The algorithm provides a computationally efficient method for the training of multilayer perceptrons (97).

3.2.5.2 Training a Neural Network

Neural network cannot approximate beyond the information contained in the training data. Therefore, one of the main important steps is to collect the qualitative data and transform it to an acceptable form for the network. The data is pre-processed, in order to remove any outliers, handle any missing data and scale it into the active range of the activation functions used. The steps for training MLP networks can be summarized as follows:

Scaling and Normalization : The performance of the network can be improved if inputs are scaled to the active domain of the activation functions. The input values needs be pre-processed so that its mean value, averaged over the entire training set, is close to zero, or else it is small compared to its standard deviation (98). This normalization helps the training process to ensure that the inputs to the hidden units are comparable to one another and to avoid saturation of the activation functions.

Network Architectures In neural network architecture the main degree of freedom are the total number of neurons and the number of hidden layers into which they are arranged. The number of training epochs, the run through training set, also varies according to the criteria adopted. The network with more number of neurons and hidden layers clearly have more degree of freedom, but a problem known as 'overfitting' of the training examples might occurs. Also when the behaviour of the data is well known, it is really advantageous to use more than one layer, like that in engine performance data (99). A proper selection of the activation function is also important. *Sigmoid* or *tanh* activation functions are the most popular choices for the units in the hidden layer while the linear activation function for the output unit is appropriate for regression problems.

Initialising Weights : The gradient-based optimisation methods is very sensitive to the initial weight vectors (100), and it is very important that the network weights and biases are initialised before the training begin. This process takes a network object as input and returns a network object with all weights and biases initialized. Random weights centred around 0 is generally chosen as a good choice for weight initialisation strategy (101). Wessels and Barnard (102) showed these random weights in the range $[-r, r]$, the range r is defined by

$$r = \frac{1}{\sqrt{N}} \tag{3.16}$$

where N is the number of inputs of the particular neuron (fanin).This initialization strategy ensures that the sigmoid activation functions start in their linear regions and not in saturation, thereby improving training performance.

Training: Train the network with the algorithm of choice (e.g., backpropagation) until sufficient fitting error is achieved for the training dataset. Many practices in neural networks suggest splitting the data sets into training and testing sets, know as *cross validation*. The former is used for network training while the latter is used to stop the training when prediction error on the testing data set achieves a minimum. When the size of the data set is small, however, this approach may be unjustified. Other practices include early stopping of training.

That is, the training iteration process is stopped after a small number of iterations before the fitting errors are too small to avoid overfit. This heuristics, however, is very adhoc as it is usually difficult to determine when to stop. Some authors have suggested using a penalty function in addition to the least square criterion function to eliminate some of the units in the hidden layer or to prune some of the weights.

Outliers : Any data pattern that deviates substantially from the data distribution is known as an outlier. Outliers have a great effect on the accuracy of the network because of the large deviation from the norm. These outliers result in large errors, and consequently large weight updates.

Since a neural network is an empirical model, it highly dependent on data used for training and validation. Clearly, the accuracy of the model itself does not exceed the data been used to develop the model, and the optimization method could converge to a biased solution itself, if trained on a biased data and thus loosing the generalization. So, it is important to consider the use of available data during the network construction by following a cross validation procedure. Trained network can reproduce accurately the target output at each point in the training set.

3.2.5.3 Radial Basis Functions (RBF)

This section discuss another type of neural network, Radial Basis Functions (RBF) along with their properties, the motivations behind their use and some of their applications are mentioned. Radial Basis Functions emerged as a variant of artificial neural network in late 1980s. However, their roots are entrenched in much older pattern recognition techniques as for example potential functions, clustering, functional approximation, spline interpolation and mixture models (103).

RBF's are embedded in a two layer neural network, where each hidden node

implement a set of radial basis functions (e.g. Gaussian functions). The output nodes implement linear summation functions as in an MLP. The network training is divided into two stages: first the weights from the input to hidden layer are determined, and then the weights from the hidden to output layer. The training/learning is very fast, with very good at interpolation. Due to their nonlinear approximation properties, RBF networks are able to model complex mappings, which perceptron neural networks can only model by means of multiple intermediary layers (97).

When the output function goes exactly through all the data points is called exact interpolation. The exact interpolation of a set of N data points in a multi-dimensional space requires all the D dimensional input vectors $x^p = x_i^p : i = 1, \ldots, D$ to be mapped onto the corresponding target outputs t^p. The goal is to find a function $f(x)$ such that

$$f(x^p) = t^p \quad \forall \quad p = 1, \ldots, N \tag{3.17}$$

The radial basis function approach introduces a set of N basis functions, one for each data point q, which take the form $\phi(||x - x^q||)$ where $\phi(.)$ is some non-linear function whose form will be discussed later. Thus the q^{th} such function depends on the distance $||x - x_i||$, usually taken to be Euclidean, between x and x^q. The output of the mapping is then taken to be a linear combination of the basis functions, i.e.

$$f(x) = \sum_{q=1}^{N} w_q \phi(||x - x^q||) \tag{3.18}$$

The idea is to find the weights w_q such that the function goes through the data points. It is easy to determine equations for the weights by combining the above equations:

$$f(x^p) = \sum_{q=1}^{N} w_q \phi(||x^p - x^q||) = t^p \tag{3.19}$$

The distances $||x^p - x^q||$ between data points p and q are fixed by the training data, so

$$\Phi_{pq} = \phi(||x^p - x^q||) \tag{3.20}$$

is simply an array, or matrix, of training data dependent constant numbers, and the weights w_q are the solutions of the linear equations

$$\sum_{q=1}^{N} w_q \Phi_{pq} = t^p \tag{3.21}$$

This can be written in matrix form by defining the vectors $t = t^p$ and $w = w_q$, and the matrix $\Phi = \phi_{pq}$, so the equation for w simplifies to $\Phi w = t$.

Determining the Weights It then follows that, provided the inverse of the matrix Φ exists, any standard matrix inversion technique can be used to give the required weights:

$$w = \Phi^{-1} t \tag{3.22}$$

where the inverse matrix Φ^{-1} is defined by $\Phi^{-1\Phi=I}$. It can be shown that, for a large class of basis functions $\phi(.)$, the matrix Φ is indeed non-singular (and hence invertible) providing the data points are distinct.

Once the weights are determined, the function $f(x)$ represents a continuous differentiable surface that passes exactly through each data point.

Commonly Used Radial Basis Functions A range of theoretical and empirical studies have indicated that many properties of the interpolating function are relatively insensitive to the precise form of the basis functions $\phi(.)$. Some of the most commonly used basis functions are:

1. Gaussian Functions:

$$\phi(r) = exp\left(-\frac{r}{2\sigma^2}\right) \qquad \text{width parameter } \sigma > 0 \tag{3.23}$$

2. Multi-Quadratic Functions:

$$\phi(r) = (r^2 + \sigma^2)^{(1/2)} \qquad \text{parameter } \sigma > 0 \tag{3.24}$$

3. Generalized Multi-Quadratic Functions:

$$\phi(r) = (r^2 + \sigma^2)^{(\beta)} \qquad \text{parameter } \sigma > 0, 1 > \beta > 0 \tag{3.25}$$

4. Inverse Multi-Quadratic Functions:

$$\phi(r) = (r^2 + \sigma^2)^{(-1/2)} \qquad \text{parameter } \sigma > 0 \tag{3.26}$$

5. Generalized Inverse Multi-Quadratic Functions:

$$\phi(r) = (r^2 + \sigma^2)^{(-\alpha)} \qquad \text{parameter } \sigma > 0, \alpha > 0 \tag{3.27}$$

6. Thin Plate Spline Function:

$$\phi(r) = r^2 ln(r) \tag{3.28}$$

7. Cubic Function:

$$\phi(r) = r^3 \tag{3.29}$$

8. Linear Function:

$$\phi(r) = r \tag{3.30}$$

Figure 3.11 shows a general architecture of a radial basis function model. In order to use RBF network the hidden unit activation function is specified, along with the number of processing units, a criterion for modelling a given task and a training algorithm for finding the parameters of the network. Finding the RBF weights is called network training. A set of input-output pairs, called training set is the optimisation of the network parameters in order to fit the network outputs to the given inputs. The fit is evaluated by means of a cost function, usually assumed to be the mean square error.

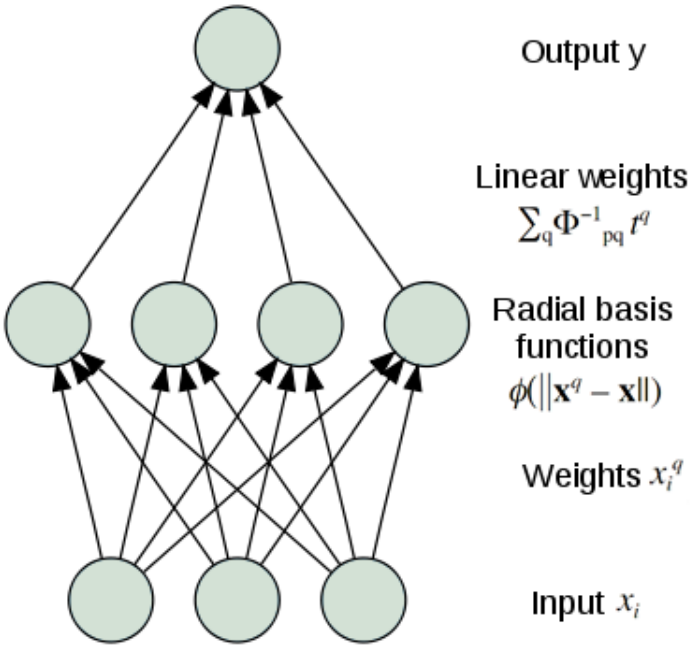


Figure 3.11: Radial Basis Function (RBF) Network

The N training patterns x_i^p, t^p determine the weights directly. The hidden layer to output weights multiply the hidden unit activations in the conventional manner, but the input to hidden layer weights are used in a very different fashion.

Training the RBF Network The steps to build RBF networks are as follows:

1. Normalise the data such that $u_i = (x_i - \bar{x}_i)/s_i$, where $\bar{x}_i$ and s_i are the mean and standard deviation of x_i, respectively. This normalisation helps

the training process to make the input scales comparable to each other; thus equal ϕ for all inputs can be applied.

2. Apply a line search algorithm to find ϕ that maximises the penalised likelihood criterion.

3. Apply least squares (or penalised least squares) to estimate w.

Problems with RBF Network It is possible to setup an RBF network for exact interpolation, but following are the two main problems with this network

1. As for Multi-Layer Perceptrons (MLPs), it is usually not intended that the network outputs pass through all the data points when the data is noisy, because that will be a highly oscillatory function that will not provide good generalization.

2. The network requires one hidden unit (i.e. one basis function) for each training data pattern, and so for large data sets the network will become very costly to evaluate. With MLPs the generalization can be improve by using more training data the opposite happens in RBF networks, and they take longer to compute as well.

However, different methods are used to improve the generalisation capability and reduction in the computational time (97).

3.2.5.4 Network Generalization

Neural network are non-parametric models that can approximate any continuous non-linear input-output relationship (104). The quality of the approximation depends on the architecture of the network used, and on the complexity of the target relationship. The problem of finding a suitable set of parameters that approximate an unknown relationship is solved usually using supervised learning algorithms, that require set of input-output training set related through a relation (105).

Learning the training set is often posed as an optimization problem by introducing an error measure. This error is a function of the training examples as well as of the network parameters, and it measures the quality of the network's approximation to the relation on the restricted domain covered by the training set. The minimization of this error over the network's parameter space is called the training process. The task of learning, however, is to minimise that error for all possible examples related through function, namely, to generalize.

Generalization is a very important aspect of neural network learning. It is the measure of the network capability to interpolate to points not used during training, to produce a learner with low generalization error. Mathematically, the goal of the network training can be formulated as minimization of a true risk function (106):

$$E_{true} = \int_{x,y} e(f(x, W), y)p(x, y)\ dxdy \tag{3.31}$$

where e is a local cost function, f is the function implemented by the network, x is the input and y is the desired output vectors of the model, and p represent the probability distribution. The objective is to optimize the weight W of the network such that the generalization error (106) E_{true} is minimized:

$$\hat{w} = argmin_w \int_{x,y} e(f(x, W), y)p(x, y)\ dxdy \tag{3.32}$$

E_{true} is the expected performance of the network on new patterns randomly chosen from $p(x, y)$. In practice $p(x, y)$ is not known. Instead, a training set $\tau = \{x_p, y_p\}_1^{N_p}$ is given, where N_p is the number of patterns, and an approximation of E_{true} is minimized, called as *training error* (106):

$$E = \sum_{p=1}^{N_p} e(x_p, y_p) \tag{3.33}$$

The neural network training consist of finding a parameter vector, the weights and

bias through a learning procedure where the training error is minimize through a cost function:

$$E = \frac{1}{2N_p} \sum_{p=1}^{N} \sum_{q=1}^{Q} (f(x, W)_q^p - y_q^p)^2 \tag{3.34}$$

where y and $f(x, W)$ are the P-dimensional measured and the Q-dimensional estimated output respectively. Also, the root mean square error (RMSE) can also be used,

$$E = \frac{1}{2N_p} \sqrt{\sum_{p=1}^{N} \sum_{q=1}^{Q} (f(x, W)_q^p - y_q^p)^2} \tag{3.35}$$

3.2.5.5 Prediction Error Evaluation

The aim of network learning is to learn the examples presented training set well, while still providing good generalization to examples not included in the training set. However, it is possible that a network exhibits a very low MSE, but had a bad generalization E_{true} due to overfitting of the training patterns. That is, the network that overfit cannot predict correct output for data patterns not seen during training. The network generalization can be improved by using a network that is just large enough to provide an adequate fit. The larger network you use, the more complex the functions the network can create, and hence too many weights (*free parameters*) due to too many hidden units and irrelevant input units. If a small enough network is used, it will not have enough power to overfit the data. Overfitting can be overcome by optimizing the network architecture and using enough training patterns.

One of the other accuracy measurements it to calculate the correlation between the output and target values for all patterns, referred to as the correlation coefficient R:

$$
\begin{aligned}
R &= \frac{1}{\sigma_x \sigma_y}\left[\sum_{i=1}^{n}(x_i-\bar{x})\sum_{i=1}^{n}(y_i-\bar{y})\right] \\
&= \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2}\sqrt{\sum_{i=1}^{n} y_i^2 - \frac{1}{n}(\sum_{i=1}^{n} y_i)^2}}
\end{aligned} \tag{3.36}
$$

where $\bar{x}$ and $\bar{y}$ are respectively the averages over all observations x_i and y_i, σ_x and σ_y are the standard deviations of the x_i and y_i observations respectively. The correlation coefficient quantifies the linear relationship between the approximated function and the true function. A correlation value close to 1 indicates a good approximation to the true function. Also, the slope m and the y-intercept b of the best linear regression relating approximated to true outputs, is calculated for the post regression analysis. If there were a perfect fit (outputs exactly equal to approximated), the slope would be 1, and the y-intercept would be 0.

Generalization is the most important aspect that has to be considered when designing a neural network. The generalization of the network is not just the measured accuracy achieved by it, but aspects such as computational complexity and convergence characteristics are just as important.

3.2.6 Comparison of RBF Network and Multilayer Perceptrons

Both Radial basis function (RBF) network and multilayer perceptrons are non-linear layered feedforward networks, and are universal approximators. An RBF network can mimic accurately a specified MLP, or vice verse. However, these two differ from each other in several important aspects (97), these include;

1. An RBF network has a single layer in its most basic form, whereas as MLP may have one or more hidden layers.

2. The computational nodes in the hidden layer of an RBF network are quite different and serve a different purpose from those in the output layer of the

network, while the computational nodes of an MLP, located in a hidden or an output layer, share a common neuronal model.

3. The hidden layer of an RBF network is nonlinear, whereas the output layer is linear. However, the hidden and output layers of an MLP used as a pattern classifier are usually all nonlinear. Although, a linear layer for the output is usually the preferred choice, when MLP is used as a pattern regression problems.

4. The argument of the activation function of each hidden unit in an RBF network computes the *Euclidean norm (distance)* between the input vector and the centre of that unit. Meanwhile, the activation function of each hidden unit in an MLP computes the *inner product* of the input vector and the synaptic weight vector of that unit.

5. An RBF networks using exponentially decaying localized non-linearities (e.g., Gaussian functions) construct *local* approximations to nonlinear input-output mappings, while MLP network construct *global* approximations to nonlinear input-output mapping.

Hence, for the approximation of a nonlinear input-output mapping, the MLP may require a smaller number of parameters then the RBF network for the same degree of accuracy.

3.2.7 Recommendations for Model Choice and Use

Simpson, Peplinski, Koch and Allen (60) and Chen, Tsui, Barton and Meckesheimer (107) presented a survey and conducted some empirical comparisons of modelling techniques, including response surface, neural networks and kriging. They recommended the following:

1. Polynomial model is linear models and is typically small and form specified by user. It is relatively very fast to run and available in any statistical package. The main disadvantage of this type of model is that it is not flexible.

2. MARS is a data-adaptive linear model and has got moderate size, because it only includes important effects. The major advantage appears to be accuracy and major reduction in computational cost associated with constructing the meta-model.

3. Kriging can model complex structure. It can predict exact at observed values, although the assumptions are difficult to verify. Model size is large and requires storage of all data points, and due to this also the estimation of model parameters is computationally intensive.

4. Induction learning is best when factors and responses are discrete-valued. It is better suited to diagnosis than engineering design.

5. Neural Networks is good for highly nonlinear or very large problem ($\sim 10,000$ parameters). It may be the best choice (despite of its high computational expense) in the presence of many factors to be modelled in a deterministic application.

6. Radial Basis function (RBF) is linear model and can model complex structure. The model is large and it includes a term for each data point. But it is fast to moderate, and also it is easy to code in Matlab.

3.3 Two-stage Regression

Lindstrom and Bates (108) define repeated measurements as data generated by observing a number of individuals repeatedly under various experimental conditions, where the individuals are assumed to constitute a random sample from a population of interest. An important class of repeated measurements is longitudinal data where the observations are taken on each of a number of subjects over time or position in space. More generally, longitudinal data is defined as repeated measurements where the observations on a single individual are not, or cannot be, randomly assigned to the levels of a treatment of interest.

Holliday (48; 49) introduced a two-stage designed model to engine mapping process, reducing the problem arising from incorrect choice of domain to the extension of model for inclusion of more adjustable engine parameters. The fact that the engine data for the mapping process are organized in sweeps and the empirical modelling benefits more from the two-stage of the data.

The focus of Holliday's work was the modelling of data taken from engine mapping experiments. In these experiments, engine speed, load, and air/fuel ratio were held constant while spark was varied. Various engine response characteristics, for example, torque or emission quantities were measured at each spark setting. Holliday modelled the response characteristics for each sweep as a function of spark advance. Variations in the individual sweep parameters were then modelled as a function of the global engine operating variables speed, load, and air/fuel ratio. Conceptually, variations in the measurements taken within a sweep represent the *intra-individual* component of variance. Similarly, variation in the sweep-specific parameters between sweeps represents the *inter-individual* component of variance. The principles can generalise engine modelling exercises where the nature of data collection usually involves sweeping a single engine control variable while the remainder are held at fixed values. These points suggest that nonlinear repeated measurements analysis represents a general approach to the parametrisation of mean value engines models for controls-oriented development.

Another application for models of this form is the flow equations for a throttle body. Assuming the flow equations are based upon the usual one-dimensional isentropic flow principle, then they must be modified by an effective area term, Ae, which accounts for the fact that the true flow is multidimensional and irreversible. The throttle flow characteristics can be mapped by sweeping the throttle position at fixed engine speed. This data collection methodology naturally imposes a hierarchy the analysis of which is consistent with the application of nonlinear repeated measures. Experience in modelling effective area suggests that free knot spline or biological growth models provide good local predictions (58). The global phase of the modelling procedure is concerned with predicting the systematic variation in the response features across engine speed. A free knot

spline model has proven useful for this purpose.

Holliday's approach contrast distinctly with that taken by many other authors (52; 57; 109; 110; 111), which does not account for the hierarchical nature of the data. Such model in general use ordinary or weighted least square regression methods for fitting. For models the experimental observations, under the standard distribution assumptions applied in regression analysis, must be uncorrelated. However, it is very likely that the neighbouring points within the same sweep will indeed be correlated if a single model is fitted to the entire data set using least squares. If these *intra-unit correlations* are large then the potential consequences for the analysis can be severe. The model constructed using two-stage method had a high level of prediction over a wide range of engine domain and had a performance considered better than a large polynomial based approach having data equals three times more than it (50).These models allow for the special correlation structure within the data by adding supplementary random factors.

3.3.1 Non-linear Model for Repeated Measurement Data

Non-linear repeated measurement data have been divided into two broad classes of inferential procedure. In case where sufficient measurements are available on individual sweeps to allow construction of sweep-specific regression coefficients, that provides the foundation of inferential procedures at stage-2 in the model, are referred as *two-stage regression* approach (48; 49**?**). However, if sufficient observations per sweep are not available the analyst has recourse to methods based on *linearisation*.

For the purpose of this research only the first case is considered, where sufficient data are available for all, or most, of the individual sweeps to permit estimation of sweep-specific regression parameters. Modeling data of this kind usually involves the characterization of the relationship between the measured response, y, and the repeated measurement factor, or covariate x. In many applications, the underlying systematic relationship between y and x is nonlinear. In some

cases the relevant nonlinear model can be derived on physical or mechanistic grounds. In other contexts however, a nonlinear relationship might be used simply to provide a convenient empirical description for the data. The presence of repeated observations on an individual requires particular care in characterizing the variation in the experimental data. In particular, it is important to represent two sources of variation explicitly: random variation among measurements *within* a given individual *(intra-individual)* and random variation *among* individuals *(inter-individual)*. Inferential procedures accommodate these different variance components within the framework of an appropriate hierarchical statistical model. This is the fundamental idea behind the analysis of repeated measurement data.

In following sections, the basic hierarchical nonlinear model defined by Davidian and Giltinan (55; 112) is discussed. The model involves two stages, with each stage considered in detail in Sections 3.3.1.1 and 3.3.1.2

Let y_{ij} denote the j^{th} response, $j = 1, ..., n_i$, for the i^{th} individual, $i = 1, ..., m$, taken at a set of conditions summarized by the vector of covariates x_{ij}, so that a total of $N = \sum_{i=1}^{m} n_i$ responses have been observed. Suppose that a nonlinear function f may be specified to model the relationship between y and x, where β is a $(p \times 1)$ vector of parameters. Although, the form of f is common to all individuals, the parameter β may vary across individuals. This possibility is taken into account by specification of a separate $(p \times 1)$ vector of parameters β_i for the i^{th} individual. The mean response for individual i depends on the regression parameter β_i specific to the individual. This may be written as

$$E(y_{ij}|\beta) = f(x_{ij}, \beta_i) \tag{3.37}$$

The two stage model is defined as follow:

3.3.1.1 Intra-sweep Variation

For individual i, the j^{th} response follows the model

$$y_{ij} = f(x_{ij}, \beta_i) + e_{ij} \tag{3.38}$$

where e_{ij} is a random error term reflecting uncertainty in the response, given the i^{th} individual, with $E(e_{ij}|\beta_i) = 0$. Collecting the n_i responses and errors for the i^{th} individual into the $(n_i \times 1)$ vectors $y_i = [y_{i1}, ..., y_{in_i}]'$ and $e_i = [e_{i1}, ..., e_{in_i}]'$, respectively. Similarly, defining the f_i vector as

$$f_i(\beta_i) = \begin{bmatrix} f(x_{i1}, \beta_i) \\ \cdot \\ \cdot \\ f(x_{in_i}, \beta_i) \end{bmatrix},$$

Hence, the model for the i^{th} individual is summarize as

$$y_i = f_i(\beta_i) + e_i, \tag{3.39}$$

where $E(e_i|\beta_i) = 0$.

The model given in Equation 3.38 and 3.39 describe the systematic and random variation associated with measurements on the i^{th} individual. Systematic variation is characterized through the regression function f, while random variation is represented by an assumption on the random errors e_i. Hence, the specification of a model for the distribution of the e_i completes the description of intra-individual variation for the i^{th} individual.

For a given individual, variability in the y_{ij} may be a systematic function of the mean response for that individual, other known constants, and additional, possibly unknown parameters. Correlation among measurements on a given individual may also arise. Thus, the random individual variation represented by the

error e_i account for heterogeneous variance within-individual as well as correlation within-individual.

For intra-individual variation, as assumption is made about the conditional distribution of e_i given β_i. The most common distributional assumption is that of intra-individual normality of the response, which follows the error specification:

$$e_i|\beta_i \sim \mathbf{N}[0, \mathbf{R}_i(\beta_i, \zeta)] \tag{3.40}$$

where $\mathbf{R}_i \in \mathbf{R}^{n_i \times n_i}$ is a covariance matrix, and $\zeta_i \in \mathbf{R}^q$ a vector of dispersion parameters that may be chosen to reflect the heterogeneity of variance, within-individual correlation, or both. The model of this type is more flexible, which allow dependence on i to be through the individual-specific information and individual mean response, given β_i

3.3.1.2 Inter-sweep Variation

To account for inter-individual variation among different sweeps, the standard approach is to specify a model for the β_i. The degree of complexity of this model will depend on the nature of the data. The general form for a model for inter-individual variation as a function of fixed parameters, individual specific characteristics, and random effects is given by

$$\beta_i = \mathbf{d}(\mathbf{a}_i, \theta, \gamma_i) \tag{3.41}$$

where $\mathbf{d}$ is a p-dimensional vector-valued function, $\theta \in \mathbf{R}^r$ is a vector of fixed parameters, $\mathbf{a}_i$ is an $(a \times 1)$ covariate vector corresponding to individual attributes for individual i and γ_i is a vector of random effects associated with the i^{th} individual.

Although, Equation 3.41 is extremely general in nature, in most cases sufficient observations per sweep exist to allow construction of sweep-specific regression parameter estimates. These may be subsequently used as building blocks for further inference and for the basis of *two-stage regression* analysis procedures.

When sufficient observations exist to permit estimation of the sweep-specific parameters, the following model at second stage is assumed (58);

$$\beta_i = \mathbf{d}(\mathbf{a}_i, \theta) + \gamma_i \tag{3.42}$$

The curve fit parameters do not usually have any intuitive interpretation from an engineering perspective, rather the characteristic geometric features of the curve are of interest. The terminology *"response features"* of Crowder and Hand (56) is used to describe these geometric features of interest. In general, the response features will be related to the fit parameters through a non-linear vector valued function, $p_i(\beta_i)$ say. Thus, the global model is concerned with relating the systematic variation in the $p_i(\beta_i)$ to changes in the remaining parameters. Therefore, the response feature vector p_i for the i^{th} sweep is a nonlinear function g of the corresponding curve fit parameter vector θ, such that:

$$p_i = g(\theta_i) \tag{3.43}$$

Modelling the variation in the response features as a function of the global variables. The response features are carried through to the second stage of the modelling procedure rather than the curve fit parameters because they have an engineering interpretation. This ensures that the second stage of the modelling process remains relatively intuitive. The global relationship between the response features and the other parameters can be approximated by a linear model with additive error. The model can be represented as

$$p_i = X_i\beta + \gamma_i, \quad i = 1, \ldots, n \tag{3.44}$$

where X_i contains the information about the covariates at which the i^{th} individual is observed, β is the vector of global parameter estimates that must be estimated by the fitting procedure, and γ_i is a vector of normally distributed random errors. It is necessary to make some assumption about the error distribution γ, and this

is typically a normal distribution with

$$\gamma_i \sim \mathcal{N}(0, \mathcal{D}) \tag{3.45}$$

where r is the number of response features. The dimensions of $\mathcal{D}$ are $(r \times r)$ and, being a variance-covariance matrix, $\mathcal{D}$ is both symmetric and positive definite. Terms on the leading diagonal of $\mathcal{D}$ represent the test-to-test variance associated with the estimate of the individual response features. Off-diagonal terms represent the covariance between pairs of response features. The estimation of these additional covariance terms in a multivariate analysis improves the precision of the parameter estimates.

The least square method is used to estimate the coefficient of both local and global models. This method chooses β's so that the sum of squares of the errors is minimized. The least square method for Equation 3.44 can be explained as follows;

In general p is a $(n \times 1)$ vector of response feature, X is a $(n \times r)$ matrix of parameters, β is a $(r \times 1)$ vector of coefficient, and γ is a $(n \times 1)$ vector of error. Then,

$$L = \sum_{i=1}^{n} \gamma_i^2 = \gamma' \times \gamma = (p - X\beta)' \times (p - X\beta) \tag{3.46}$$

where L is square of errors

$$\begin{aligned} L &= p' \times p - \beta' X' p - \beta' X' p + \beta' X' X \beta \\ &= p' \times p - 2\beta' X' p + \beta' X' X \beta \end{aligned} \tag{3.47}$$

Since $\beta' X' p$ is a (1×1) matrix, or a scalar, and its transpose $(\beta' X' p)' = p' X \beta$ is

the same scalar (**?**). The least square method must satisfy

$$\frac{\partial L}{\partial \beta}\hat{\beta} = -2X'p + 2X'X\hat{\beta} = 0 \rightarrow \hat{\beta} = (X'X)^{-1}X'p \tag{3.48}$$

Therefore, the fitted model is

$$\hat{p} = X\hat{\beta} \tag{3.49}$$

3.3.1.3 Selection Criteria

Cross-validation, sometimes called rotation estimation,(113) is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. Note that the validation set must be distinct from the training set for the assessed performance to be valid.

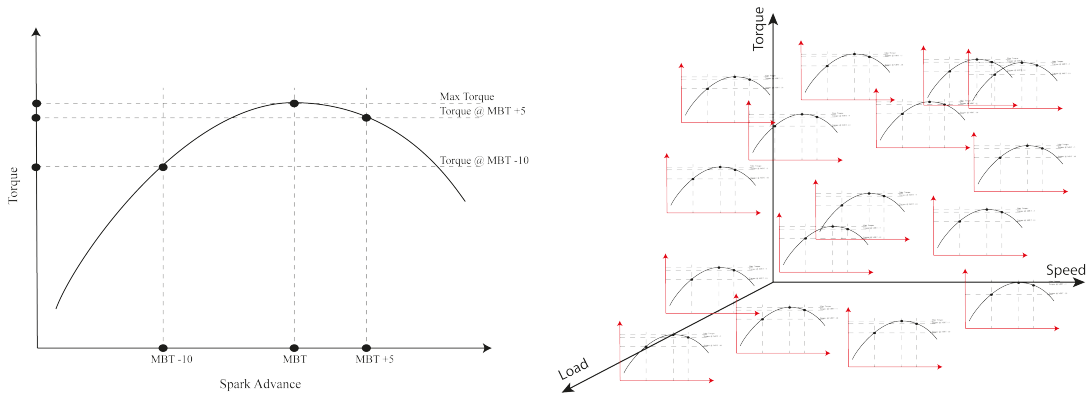
Leave-one-out (LOO) cross-validation (114) involves using a single observation from the original sample as the validation data, and the remaining observations as the training data. This is repeated such that each observation in the sample is used once as the validation data. Leave-one-out cross-validation is usually very expensive from a computational point of view because of the large number of times the training process is repeated. However, it possesses the advantage that all the data can be used for training, none has to be held back for validation purposes.

Generalized cross-validation (GCV) is just one of a number of criteria that all involve an adjustment to the average mean-squared-error over the training set.

The justification for GCV as a model selection criterion was first provided by (115). Orr (114) consider Bayesian information criterion (BIC). Riply (116) discusses the use of Akaike's information criterion (AIC) for selecting neural network architectures. If the number of parameters is large compared to the number of observations, the AIC may perform poorly (117). However, Burnham and Anderson (118) advocate the use of a second order variant of AIC termed AIC_c when the ratio of the sample size to the number of parameters is less than 40(117).

3.3.2 Construction of Two-Stage Regression Model

Intra sweep variations are modelled in Local models to find the best fit of a curve to the data in each test. Consider a case, where the test is a sweep of torque against spark angle, with speed, load, and AFR held at a constant value for each sweep. The following Figure 3.12(a) illustrates a single sweep with a local model fitted.

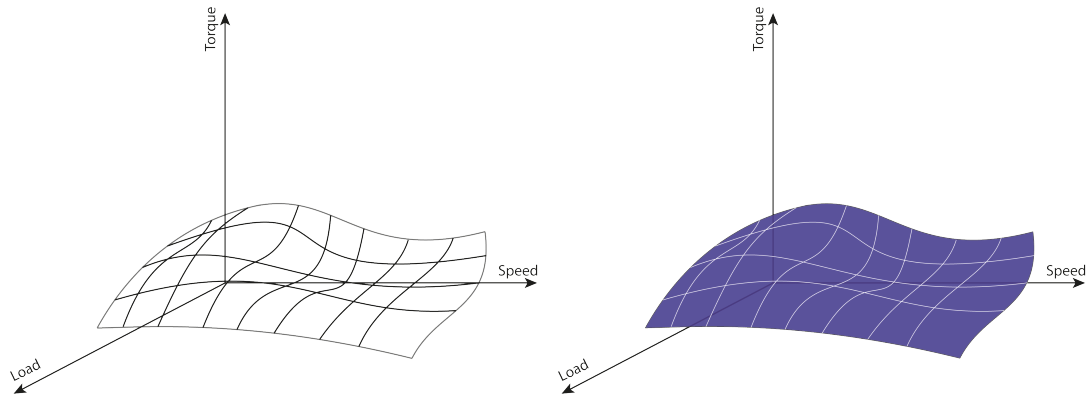


(a) Local model fit to data **(b)** Test plan in Speed/Load/AFR Space

Figure 3.12: Local model fitting in Speed/Load/AFR Space

These local models provide the coefficients to accommodate inter sweep variation in global models. The equations describing those local model curves have certain coefficients such as peak torque (PkTQ) and MBT spark (MBT; the spark angle that generates maximum brake torque).

In the development of two-stage regression models, local models are fitted to each test, in different places across the global space, as illustrated in the Figure 3.12(b). The coefficients of local models i.e. MBT and PkTQ, become the data to which the global models are fitted. These coefficients are used to make the second stage of modelling more intuitive; and have a much better understanding of a feature such as MBT spark varies through the global factor space than some esoteric curve fit parameter. These variables are helpful to engineers trying to decide how well a model describes engine behaviour. Better intuitive understanding allows much greater confidence in the models.



(a) Global model in Speed/Load/AFR Space

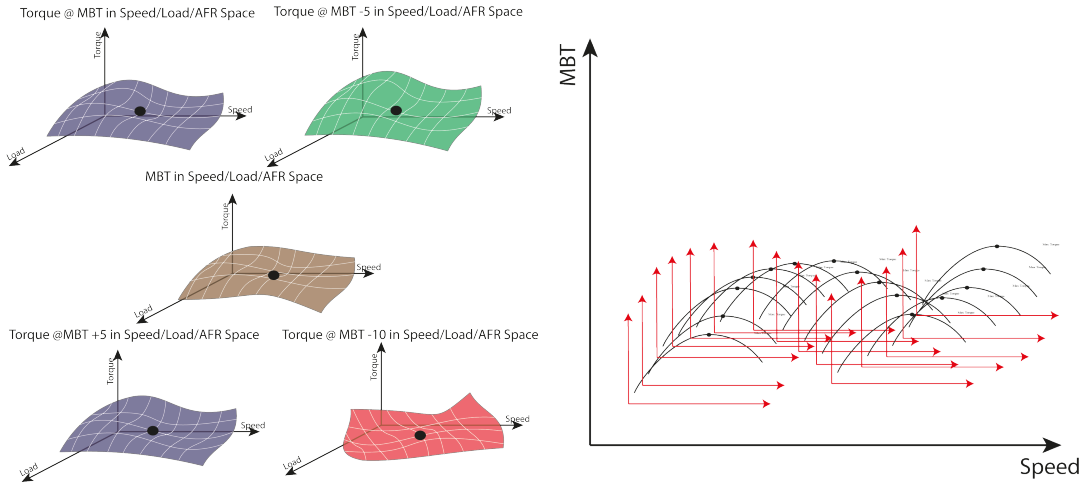
(b) Surface fitted to global model

Figure 3.13: Global model fitting

Global models are the best fit of a curve to the values of, for example, MBT for each test. This is repeated for each coefficient, producing several global models fitted to different coefficients of the local models. These coefficients are referred to as response features of the local models. The Figure 3.13(a) shows a global model for maximum torque across the speed/load global space.

The two-stage model is a surface fitted across all the global models, shown in Figure 3.13(b), to describe the behaviour across all global variables.

It can be useful to think of local and global models as a series of 2-D slices, while the two-stage model fits a 3-D surface across the curves of the global model slices. It is difficult to visualize more dimensions. Figure 3.14(a) illustrate a variety of 3-D plots of global models for properties of the local torque/spark curves (such as MBT, peak torque, and torque that is number of degrees before and after MBT), showing how these properties vary across the speed/load global space. The 2-D plot of the global MBT model (in Figure 3.14(b)) demonstrates how MBT varies with engine speed.



(a) 3-D Plots of global model **(b)** 2-D Plots of global model

Figure 3.14: Plots of global model

The two-stage model can take values of each coefficient at a certain value of, say, speed, to generate a new curve of torque against spark. This is a slice through the two-stage model surface.

In the end, the two-stage model can be tested by comparing it with the local fit and with the data. For example, a local torque/spark curve at an operating point can be reconstructed by taking the values of MBT and peak torque and the curvature from the two-stage model, and then validate this reconstructed

curve against the original fit and the data. The two-stage model can also predict responses between tests, for new sweeps at intermediate values for which there is no data. If the two-stage model shows an accurate fit when compared to the local sweeps, this is a good sign that the engine behaviour is well described by the model across the global variables.

3.4 Summary

The chapter presented review and analysis of different types of experimental design and modelling techniques used in model based methodology. A survey of experimental design is presented for the purpose of an effective utilisation of the system, for the generation of a representative data for model building. An empirical relationship between the input and outputs is provided with the choice of different approximating model and fitting methods. Parametric and non-parametric models for the analysis purpose are discussed in detail. The chapter also provide comparison and recommendation of different modelling methods.

Also, mathematics of two-stage regression approach for repeated measurement data was also discussed. Two sources of variation in the experimental data is represented, random variation within a given individual (intra-individual) and random variation among individuals (inter-individual). These two type of variation form the basis for two-stage model, with former represented in stage-1 (local model) and the later in stage-2 (global model).

4

The Designed Engine Experiments

This chapter describe the design of engine experiments for the development of steady state engine calibration using two-stage regression method. The options available for designing an appropriate experiment are discussed, with the discussion of the metrics for design on the ground of prediction accuracy. A careful selection of appropriate design and of input variable range are the key contributing factors to success in the model based process (119).

In the study a high-fidelity model data of Maloney (3) for a 2.2L inline 4 cylinder, naturally aspirated port fuel-injected spark ignition (SI) engine is used. The engine has dual overhead cams (DOHC) 4 valve per cylinder, throttle-less, and equipped with dual-independent variable cam-phaser (DIVCP) and continuously variable intake valve lift (CVIVL) actuators. The simulation GT-POWER engine model with predictive combustion capability is used by (3) with residual fraction constraint as a surrogate indication for engine instability, in place of covariance of indicated mean effective pressure (IMEP). The data is collected using space-filling design that do not depend on model type; and the most suitable model can be choose to construct a design, and when data is collected, a different model type can be tried that produces the best fit. This is the main reason for the util-

isation of the data in this research where different model types are constructed and compared.

The two-stage experimental design give us the data set, based on engine output variables over sweeps of spark advance values. This allow us to investigate the non-linearity between identical spark sweeps for estimation of desired features, such as maximum brake torque at certain spark advance. This approach requires fewer experiments, and yet contain more information (49) and thus can be used to fit a statistical engine model, e.g. a polynomial or neural network.

4.1 Design of Experiment Generation

As stated earlier, the two stage modelling methodology will be used. The aim is to ensure the development of optimal engine calibrations for complex engines with many controllable variables at minimum cost and time. The method is far more optimum then a separate experiment designed to calibrate a particular feature, which consume too much of the test cell time and is expensive. Therefore, it is important to design a generic experiment with enough contingencies built-in to cope with any unexpected anomalies.

Here, the intention is to select one primary feature and then to build responses for other features from the primary feature data set. Since the modelling was for a gasoline engine, the Spark/Torque relationship would be used as the primary feature. As with any experimental design it would help to choose most effective points to run to get the maximum information in the shortest time. But it is important that any need should be identified prior to starting any design. The design of experiment will help both in characterising the new systems as well as the system to complete a calibration. However, there is a vast difference in designing an experiment for these two types of systems. For example, with the former, all the variable interactions may not be understood and therefore the response surfaces will identify these such that control algorithm development can take place. While with the latter, the variable interactions have been identified

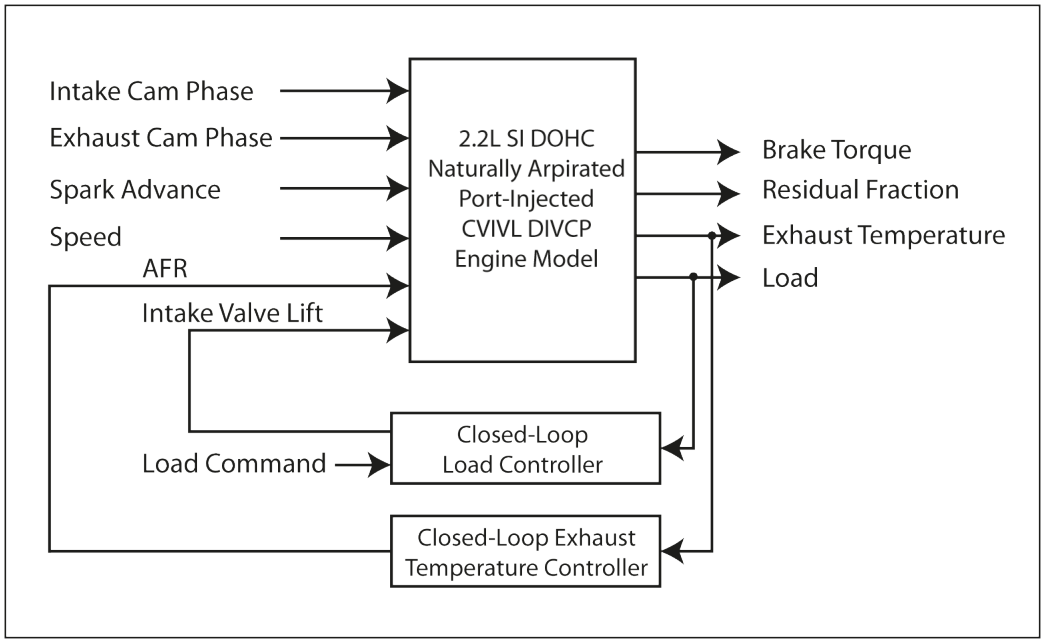


Figure 4.1: Engine test I/O Configuration

and any insignificant ones already discarded. Experimental design test points can be constrained based on previous experience to avoid damaging the engine at unrealistic operating points. For the work presented, the ranges and engineering resolutions of the test factors are shown in Table 4.1.

Table 4.1: Engine control parameter and their range

Local Inputs	Units	Symbol	Min	Max
Speed	RPM	N	500	6000
Load	%	L	0.05	0.95
Intake Cam Phase	° Crank	ICP	-5	50
Exhaust Cam Phase	° Crank	ECP	-5	50
Spark Advance	° BTDC	S	0	50

Engine testing is required to generate data necessary to determine optimal steady-

state engine calibration tables for intake cam phase (ICP), exhaust cam phase (ECP), and spark advance (S) as a function of operating point defined by engine speed. Figure 4.1 shows the test configuration of a system used.

The engine load variables here is defined as normalized cylinder fresh air mass at intake cam phase (ICP), which can be inferred by measuring engine fuel flow rate, engine speed, and engine exhaust air-to-fuel (AFR) ratio. Inferred cylinder fresh air mass is normalized to a dimensionless load value by dividing it by the cylinder air mass at piston bottom dead center (BDC), standard temperature and pressure (STP) conditions, and zero engine speed. The spark advance (S), engine speed (N), intake cam phase (ICP), exhaust cam phase (ECP), and AFR is controlled directly during the engine testing. Intake valve lift is then adjusted until a given load (LOAD) command target is reached. The AFR is adjusted down (enriched) in cases where the engine exhaust temperature is too high for the catalyst and/or engine materials, until the temperature falls to an acceptable level. However, AFR is not included explicitly as an independent variable, and non-stoichiometric operating points are taken out of the resulting data-set before analysis of the stoichiometric operating region. Identification of the input factors depends upon the engine architecture.

4.1.1 Space-Filling Design

Having discussed the utilisation of the second stage model, focus is now on corresponding experimental design procedures. The number of points required to be tested depend upon the type of model selected and will also determine to what extent the curvature of a given response can best be modelled. For example, if a response feature can be described adequately using a quadratic model, three test points are required to generate the coefficients of such a model, then it may seem pointless to test at more than three levels for each input variable. If the requirement is to improve modelling of the curvature of the response, then deviation from a quadratic model will be necessary and more than three levels are required for each factor. The information content in the available training data is considered to be the most important factor in characterizing the response at

untried input configurations of the model and the accuracy of its prediction equation. The proper experimental design increase the information content contained in the training data, sufficient to identify the character of the true underlying function.

Stevens *et al* (48) provide a quantitative example of the importance of experimental design for a study involving PFI engine. The experimental design used in his case study was a Box-Behnken (120) design for four factors (engine speed, manifold pressure (MAP), AFR and transformed spark advance factor) employed to map engine stability and combustion performance parameters. The design at three level, is intended to enable the derivation of a second order response equations from the data.

Stevens' work clearly shows the deficiency of the fitted second order equations in predicting the responses, on the validation data set at untried input configuration. A non-parametric estimator is fitted to these data - a neural network, which gives no better results than the second order polynomial models in terms of prediction errors. Stevens suggested that, it is the design rather than the data processing method that is a fault. Once the data is taken, no amount of sophisticated analysis can be added to it. And thus, the neural network trained with data designed fit a second order polynomial model behaves in a similar fashion.

For a fixed design size, Cary (58) suggested that experimental design protocols that attempt to maximize the systematic information carried by the training data should be employed as the non-parametric estimators are capable of representing a very wide class of fit functions.

A space-filling design is best for exploring a new system where prior knowledge about the underlying effects of factors and responses is low. In these type of design the available points are spread in a relatively uniform fashion on entire region to capture as much information as possible, and does not assume a particular model form. Deciding on the model to design for is vital for optimal designs only, when there is already some knowledge of the system behaviour, and it can

help to find the most efficient points for fitting the most robust models. However, space-filling design do not depend on model type; and the most suitable model can be choose to construct a design, and when data is collected, a different model type can be tried that produces the best fit.

4.1.2 The Main Design

The first step in calibration development process is to develop a unconstrained experimental design using the engine factors shown in Table 4.1. Here, the experiment designed is based on a space filling design method of Latin Hypercube Sampling (LHS) discussed in 3.1.2.2 to collect data in terms of torque/spark sweeps. LHS is an extension of stratified sampling which ensures that each of the input variables has all portions of its range represented (76; 79). The important point to consider is the number of design points for a sweep. If the design points are too sparsely distributed, then it will be impossible to sample the region with rapidly changing surface curvature. It will smooth out the response surface in such regions due to lack of sufficient data regarding its true geometry. Also, testing is expensive and time-consuming and any design that is too dense will also be costly in terms of available resources.

The process for objectively determining the cost-feasible minimum number of torque/spark sweeps to optimally calibrate a SI DIVCP is given in (3). The final design contain a total design points of 202 for four input factors, chosen to characterize the operating stage of the engine. An analysis of the process outputs shown by (3) suggested the optimal number of torque/spark sweeps required to calibrate an SI DIVCP engine, given the assumption that the model and engine architecture used are relevant to a production application of interest. The data set consist of Engine speed (RPM), Load (LOAD), Intake Cam Phase (ICP) and Exhaust Cam Phase (ECP) that are to be swept at different value of Spark Timing (S). During each sweep these four input factors are held constant while spark sweep is varied from its minimum to maximum value.

From the designed experiment, three outputs as Torque (BTQ), Exhaust Temperature (EXTEMP) and Residual Fraction (RFRAC) are recorded. This results in data from each operating point at different value of spark sweep, resulting in total of 2112 points for the main design.

4.1.3 Boundary Constraints

The experimental design is subject to the different constraints, to ensure that the engine runs in its *region of operability*, the set that will not seriously damage the engine, emissions after treatment system or test cell equipment. Therefore, it is necessary to develop a model of the operating envelope covered by the factors in Table 4.1 that can be useful when evaluating results and global models and could help in the optimal survey calibration development process. As shown in the dark area of Figure 4.2, a boundary model is fitted to the N/L factor space to account for the natural speed/load operating envelope of the engine, primarily related to the breathing capability of the engine, but also reflecting the region of operation where positive brake torque and stoichiometric operation were possible.

A local range-restriction boundary model was also fitted for spark advance S as function of the global variables N, L, ICP, and ECP using a NN model type.

These boundary models are subjected to;

- Constrain solutions to lie within the boundary constraint model (to keep the engine within its operating region),
- Constrain cam phase solutions so they do not change by more than 10° between table cells (that is, no more than 10° per 500 RPM change and per 0.1 load change), and
- Constrain residual fraction $\leq 35\%$ at each drive cycle point (to ensure stable combustion). Residual fraction is the percentage of burned gas mass in the cylinder at intake valve close, relative to the total mass in the cylinder at intake valve close. Constrain maximum residual fraction is a simple and reasonable way of ensuring stable combustion, and

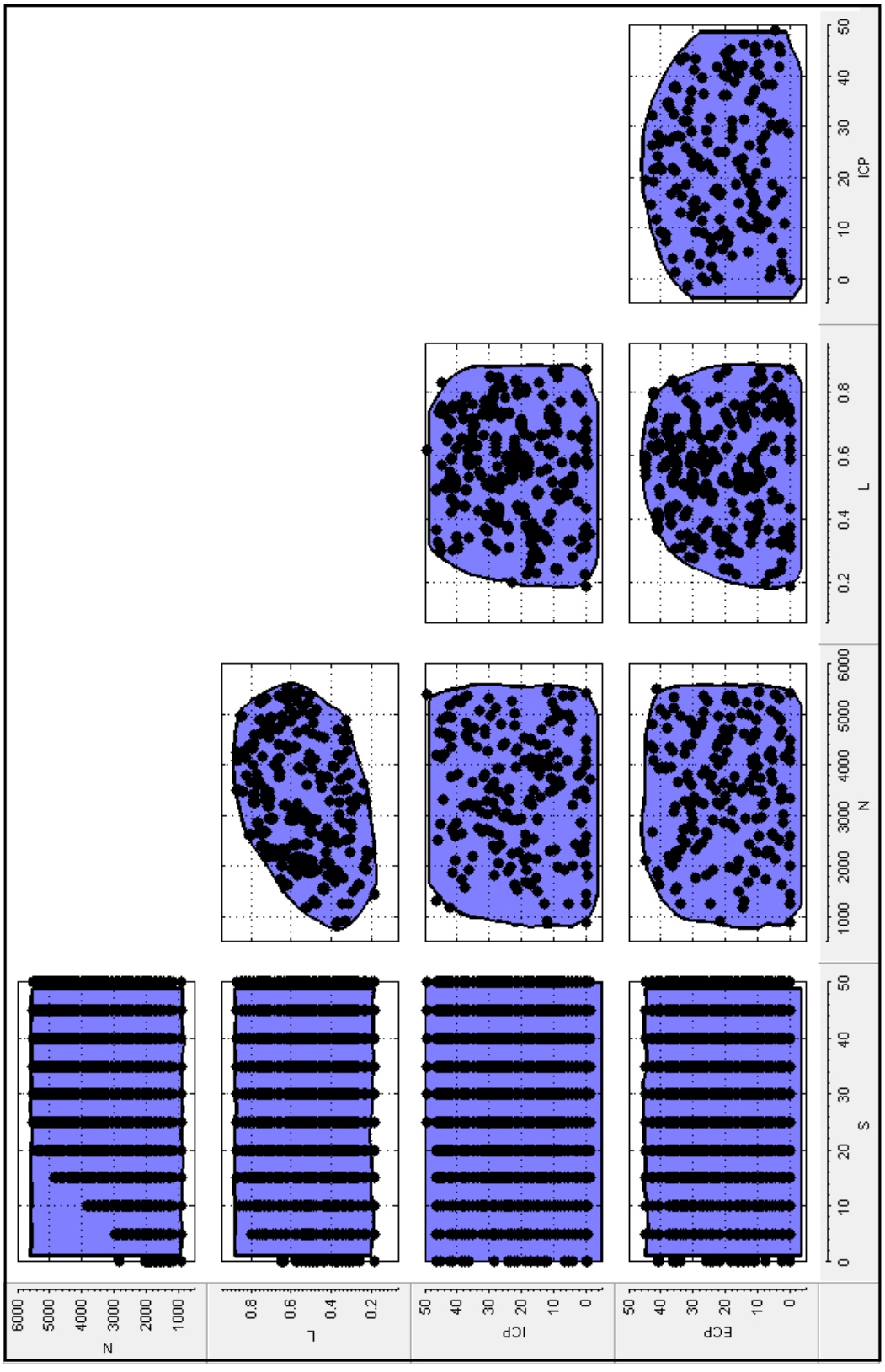


Figure 4.2: A pairwise projection of the boundary constrain

- Constrain exhaust temperature $\leq 1200°$ at each drive cycle point (to protect the catalyst).

4.1.4 The Validation Design

The model is validated to a fresh data, to assess the validity of the model prior to its use. This process of predicting fresh data is referred to as *external validation* (58). In addition, the *internal validation* proves a critical step, ensuring generally that the model behaves according with the physical theory by considerable attention paid to the model coefficients and their sign. A model is rarely acceptable if it does not have good expected physical trends, even if the external validation results of the model shows satisfactory prediction.

The process involves the collection of fresh data for the study of model's predictive performance, i.e., the model predicts the response characteristic at input configuration that has not been used for training. The validation design should be compact, but should exercise over the full range of conditions associated with its intended use.

Therefore, to assess the accuracy of the response equation maps, a validation design is also constructed on the same design method with number of arbitrary engine operation conditions. The design contain 25 points for the same four input factors for each of the spark sweep, resulting in a total of 275 points. This design is only used for the validation of the model after it has been trained on the main experimental design.

Figure 4.3 show a two-dimensional projections for the main and validation design.

4.2 Prediction Error Variance

Prediction Error Variance (PEV) is a good metric for the quality of the prediction afforded by the model. These metrics need to be defined to identify whether

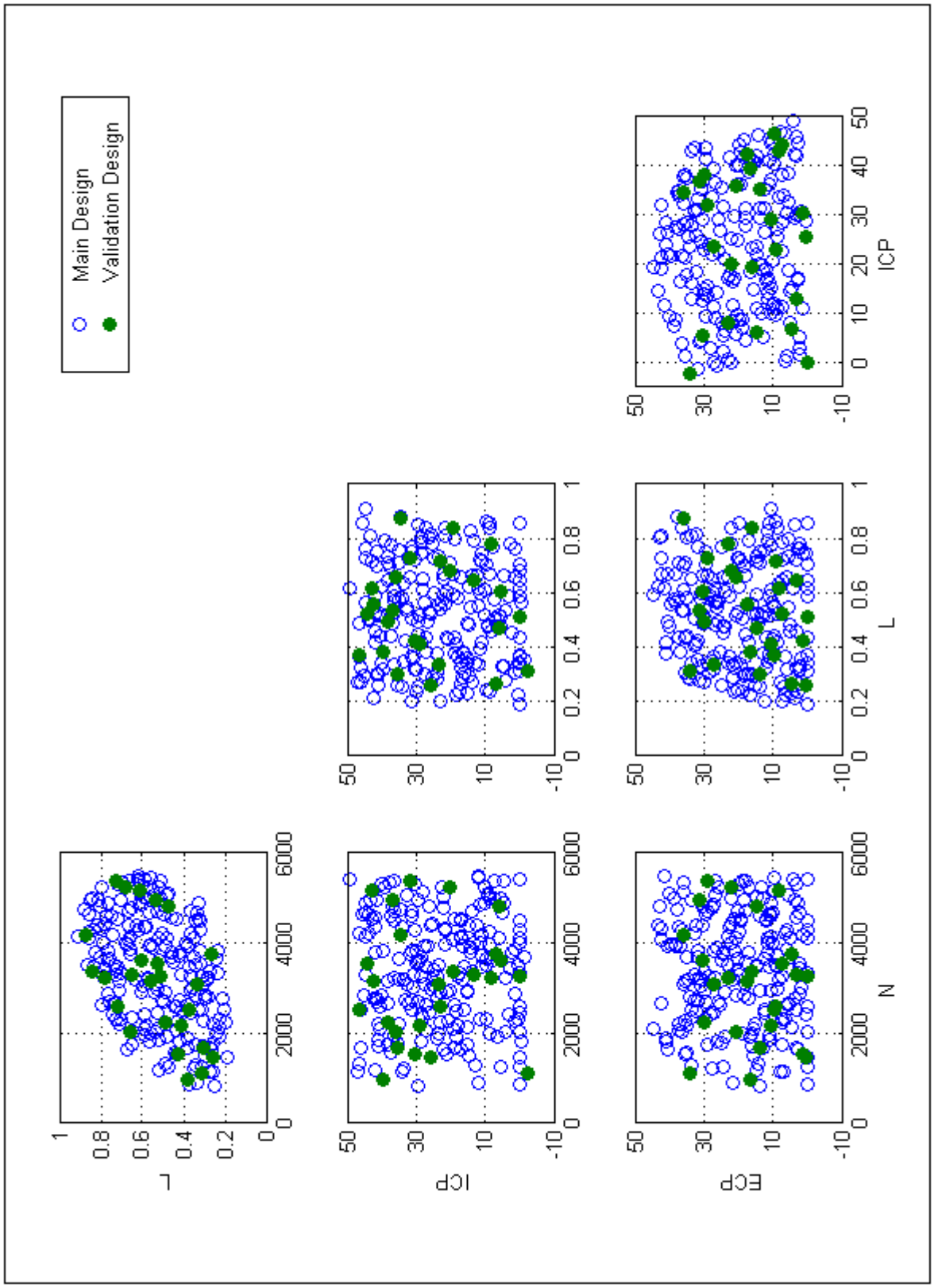


Figure 4.3: A pairwise projection of the main and validation design

the designed experiment will satisfy the calibration requirements on the grounds of prediction accuracy. A response model must be declared before so that the design can be interrogated to identify whether enough points and their locations are suitable to construct the response surface. Obviously this is impossible if the response is not known a priori, to provide an insight into the type of response some preliminary testing can be completed or alternatively an additional engine-modelling package can be used to predict the response.

Once the response model is known the model coefficients can be identified; with this information it can be shown (121) that the predicted error variance only depends upon the variance of the measurement error in the observed values (i.e. error from the predicted response surface). At this point no data has been collected, but it is still useful to observe the prediction error variance (PEV) since a number greater than 1 will magnify the measurement error and those less than 1 will reduce the error. Therefore a low PEV (tending towards zero) means that good predictions are obtained for that point.

To be able to use the PEV as a measure of success (i.e. PEV ≤ 1 within the operational area) it is therefore impossible to design an experiment without declaring or estimating the response model. In this instance it was decided to select a perhaps overly complex response model since it is possible to reduce the order of the response if required. Additionally, the use of a space filling design provides contingency in allowing a complex model such as neural network to be used, as its been discussed that these type of design the available points are spread in a relatively uniform fashion on entire region and does not assume a particular model form. A hybrid B-Spline polynomial (HBSP) model is used previously to accurately predict the MBT spark response(36; 58). Figure 4.4 shows the PEV for the design space and is less than one for the engine operation region.

One of the difficulties in using space filling designs is the fact that the point placing algorithm used, such as 'maximise the minimum distance between points', witnesses the majority of the points a significant distance from the constraints

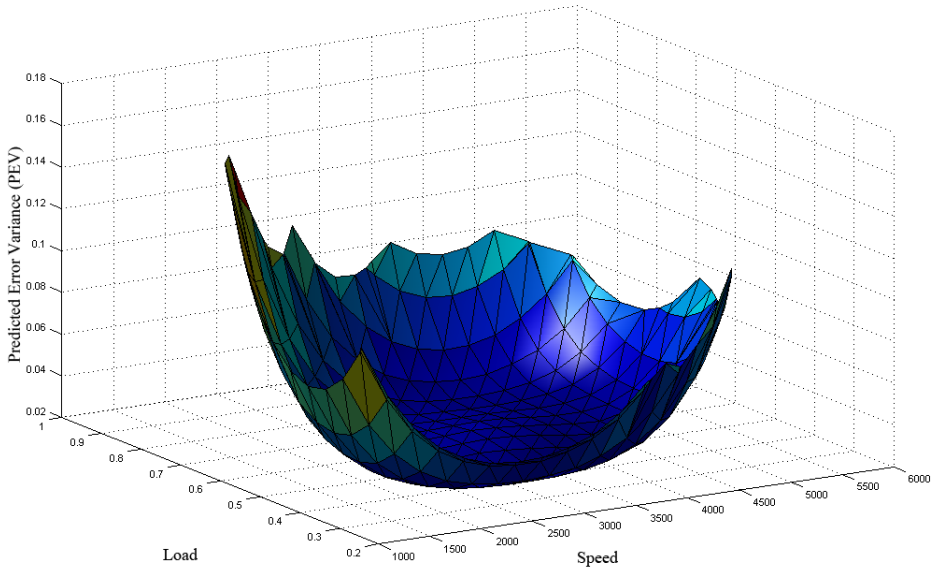


Figure 4.4: Design prediction error variance (PEV)

line or edge. Since it is difficult for the neural networks to extrapolate outside the data point region it is imperative that data points reside upon the constraints line in order to maximise the accuracy of the generated model. Points can, of course, be manually added to the design in a random fashion, but it has been found that augmenting the space filling design with an optimal design provides significant benefits. Firstly data points will be located upon the constraint line, and secondly, the algorithm will interrogate the current design and select points in areas of potential weakness.

Historically D-Optimal designs have been specified in order to provide the tightest confidence intervals on the parameter estimates. More recently V-Optimal designs have been seen as the way forward to improve the model prediction (2). Optimal designs therefore rely upon the constraints region being identified correctly. This combination of completing a space filling / optimised design process provides excellent PEV statistics but also allows non-linear neural models to be fitted to the data thereby improving the predictability of the model. Another advantage in providing a space filling design is the fact that rigidity of a classical

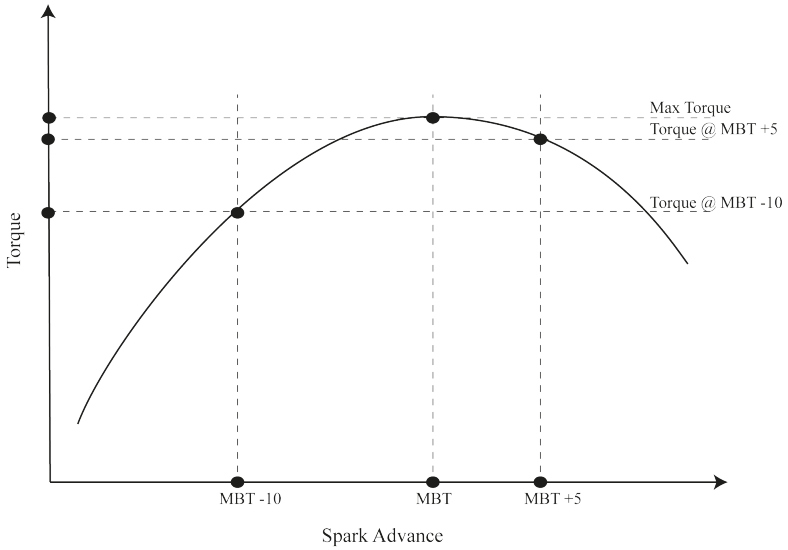


Figure 4.5: Local model characteristics

DOE is avoided, for example if the design point cannot quite be met (due to incorrect constraints identification, environmental factors or noise) the experimental design is not degraded significantly.

4.3 Sweep Definition

The local models are fitted to the data in each test. And, each test is a sweep of torque against spark angle, with speed, load, intake and exhaust cam phase held at constant value for each sweep. Figure 4.5 represents a spark sweep with a local model fitted. As spark is increased throughout its range, torque rises to a maximum, and then falls. Spark knock is not modelled, so spark advance was not limited as it would be in normal test cell, where knock-limited spark advance is typically modelled as a separate response and used later in optimisation.

The local model provide the coefficient to generate global models. The equations describing those local model curves have certain coefficients such as PKTQ and knot (abscissa values of the join points), which in this case is peak torque and MBT spark (the spark angle that generate maximum brake torque) respectively.

Two cubic polynomial will be fitted to the torque/spark curves, where different curvature is required above and below the maximum. Therefore a minimum of 7 points would be required, but increasing this to 11 generally would improves the model fit and also ensures that there are enough points past MBT to reconstruct the cubic curve.

4.4 Data Quality Checks

Before the data can be used for modelling, it should be ensure that the data is of highest level and quality. The quality of the data is the level of care taken in the design of experiment and the collection of data. If the data is of poor quality, the information content that can be extracted from it would be limited. And no amount of post-data analysis can rectify deficiencies in data quality.

Outliers have severe effects on accuracy. An outlier is a data pattern that deviates substantially from the data distribution. Because of the large deviation from the norm, outliers result in large errors, and consequently large weight updates. Figure 4.6 shows that larger differences between measured and predicted values cause an exponential increase in the error. It can be clearly seen the the fitted function is pulled toward the outliers in an attempt to reduce the training error. As result, the generalisation deteriorates and the prediction error increases.

All those data point that are unsuitable for modelling, for instance, unstable points on the edge of the engine's operating envelope, should be remove before training starts to ensure the successful implementation of any empirical modelling. However, such actions will eliminate the outlier problem, but it is believed that important information about the data might also be removed at the same time. Therefore, care should be taken while deciding about the nature of the data points, and only those point should be removed which clearly effect negatively the fit of a model.

For the data in the study, different filters are applied to remove any suspect

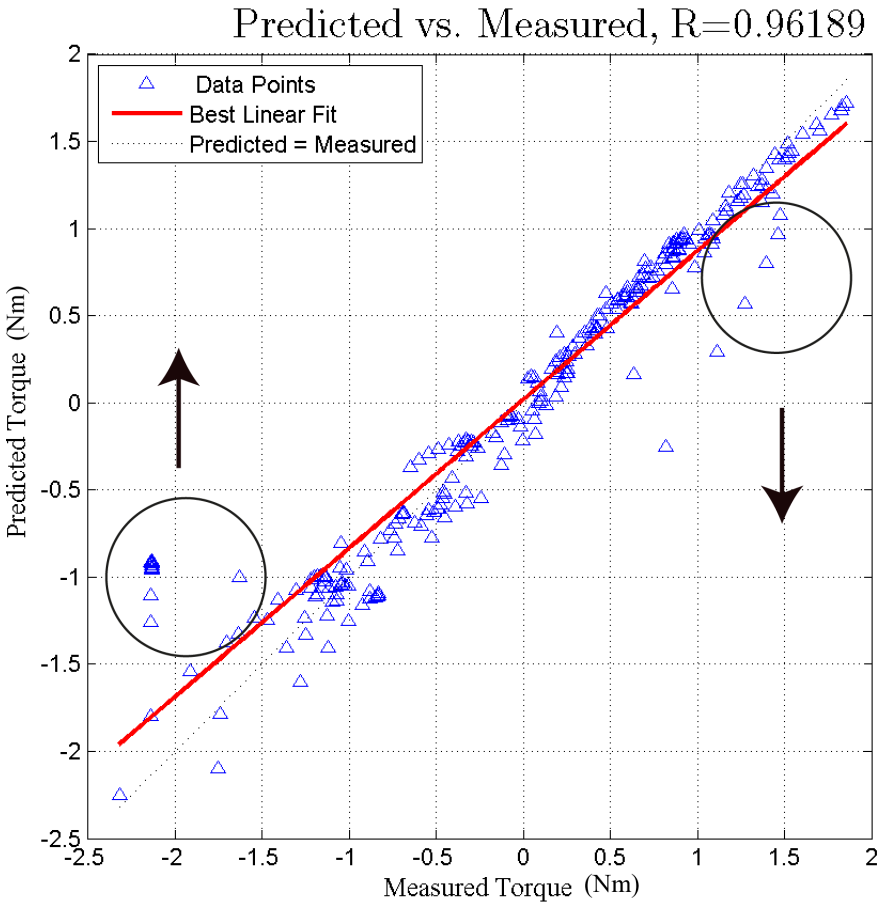


Figure 4.6: Effect of outlier on model prediction

point that is badly distorting the torque spark curve. Also, for tests where majority of the points are at higher spark angles than the maximum break torque (MBT), the fit can be improve by removing some of these long 'tails'. It can be useful to remove outliers in this region, because there is likely to be knock at spark values much higher than MBT where the engine is less stable. Similarly, as there is no knock in simulation data, points can be collected far in advance of MBT, and removing these can improve the fit.

Different other filters were also implemented in Matlab's Model Based Calibration Toolbox (121). These filters were use to keep records with AFR value greater than 14.25 to limit the exhaust temperature(AFR >12.25), residual fraction value

less than 35 to ensure stable combustion (RFRAC <35), and also to keep only those tests with sufficient points to fit the model (i.e., at least 5 points, length (BTQ) >4).

4.5 Summary

An experimental design based on the form of the two stage regression model is created, which allow the sweep based data collection employed in engine mapping process. A space-filling design based on Latin hypercube sampling technique is selected for exploring the design space for the underlying effects of factors and responses. Space-filling design is best for exploring a new system with low knowledge regarding the parameters effects, and had a significance for using in neural network based models..

The main experimental design contain a total of 202 design points for 5 input factors, with speed, load, intake cam and exhaust cam phase that are to be swept at different value of spark advance. The experimental design is subjected to different constraints to ensure that engine runs in its operability region.

Also, external validation of the model is performed on the validation design, which involves the collection of fresh data for the study of model's predictive performance.

5

Neural Network Based Engine Model

This chapter presents a steady state engine mapping case study illustrating the application of two stage regression techniques to the analysis of engine brake torque data discussed in chapter 4. To improve the predictive capability of these models a Multi-Layer Perceptron (MLP) based neural network model is introduced. At stage-2 each of the response features is modelled using MLP networks. Neural network is used to develop accurate models for the behaviour of brake torque, with additional responses as exhaust temperature and residual fraction at different values of spark advance at stage 1 and the response features as a function of speed, load and cam timings at stage-2. The engine data to build these models is collected using an experimental design on the most useful set of points discussed in the previous chapter.

As such, this chapter present some original work, with MLP used in the global model form with model structure discussed. For the purpose of comparison, Radial Basis Function (RBF) models are also fitted to the same response feature data. Model selection criteria are used to rank the models, but it is the model behaviour to the know physics of the situation which is more important.

The performance of MLP and RBF models are quite similar in terms of fit quality. However, the number of parameter increases in case of MLP with the same number of data. Hence, for the approximation of a nonlinear input-output mapping, the MLP based neural network require a smaller number of parameter than the RBF network for the same degree of accuracy. The MLP based engine model show great improvement in the prediction capability of the steady state two-stage regression approach. Therefore, the use of neural network make it possible to perform the calibration studies over a large range of operating domain, with good model fit and prediction. The model is successfully validated on a data set that was not used in the development of the model.

5.1 Neural Network Based Two Stage Engine Model

The two-stage model approach separate one or more of the input variables into a 'first stage', with the remainder falling into the 'second stage'. A multilayer perceptron network is used in the second stage of the model.

In the first step, the inputs combinations are determined with a design of experiments (DoE) described in Section 4.1.1. This involves the creation of a statistical model of the engine that can predict the output response as a function of input variables, and constrained at different level to ensure combustion stability, exhaust temperature considerations and avoiding spark knock. The behaviour of torque, exhaust temperature, and residual fraction at different values of speed, load, cam timings and spark advance are modelled. Figure 5.1 summarise the overall engine mapping in two-stage regression processes in schematic form.

In this study, the spark sweep is the fundamental experimental unit with each sweep comprising of 11 (spark, Torque) pairs, under different constraints. The spark (the local variable) is swept while the other variables, speed, load and cam timings (the global variables) are kept constant. As discussed in previous chapter, there are total of 202 designed point. These designed points are used for the

collection of the engine data, ensuring the data quality and preventing engine damage.

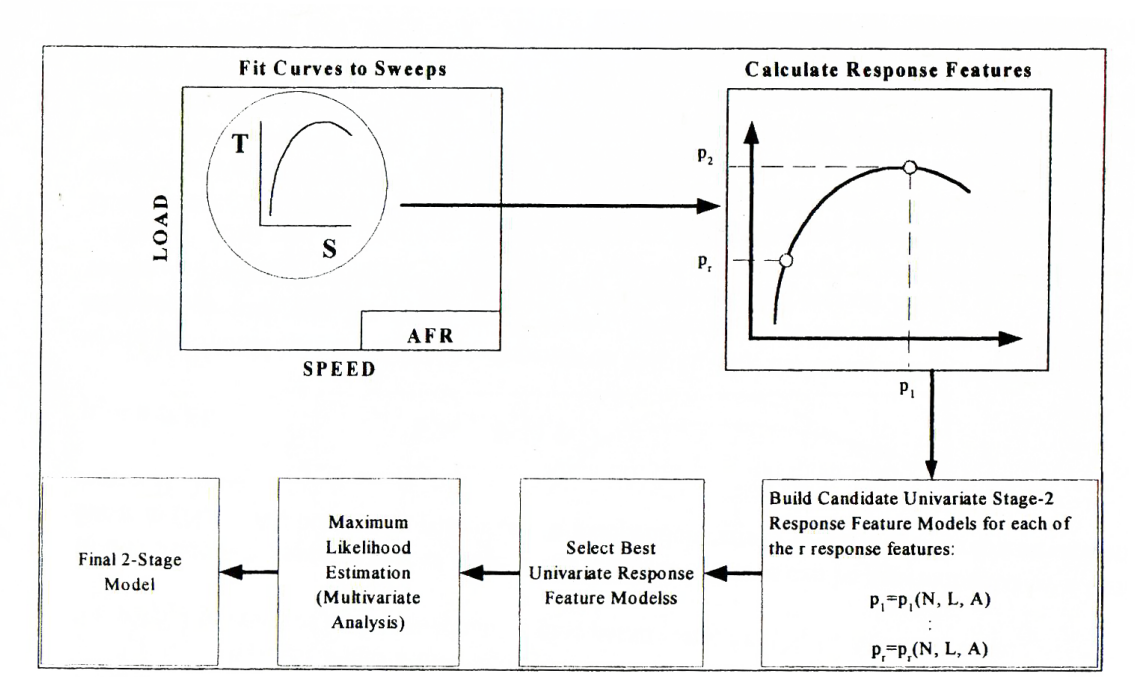


Figure 5.1: Two-stage modelling process schematic (2)

A space-filling design algorithm implemented in Mathworks' Model Based Calibration Toolbox ® was employed to generate the (N, L, ICP, ECP) test plan. The available points are spread in a relatively uniform fashion on entire region to capture as much information as possible.Deciding on the model to design for is vital for optimal designs only, when there is already some knowledge of the system behaviour, and it can help to find the most efficient points for fitting the most robust models. However, space-filling design do not depend on model type; and the most suitable model can be choose to construct a design, and when data is collected, a different model type can be tried that produces the best fit.

Modelling is divided into two-stages: The first stage *intra-sweep* and second stage

inter-sweep modelling. Intra-sweep or *Local Model* is concerned with the variation *within* a given variable or, the relationship between the response of interest and the swept variable. Each time a sweep is repeated a slightly different profile is obtained, even when the settings of the remaining experimental variables are not varied. A non-linear fit function summarise the relationship for each sweep, while the corresponding regression parameters characterise the shape of the sweep-specific response profile.

In local model, the curves are fitted using polynomial spline to individual sweep response profiles, resulting a separate model for each test group. This yields 202 separate models, one for each sweep that describe the behaviour in response as the spark advance is varied. The data diagnostic checks are conducted to remove outliers that are badly distorting the torque spark curve to improve fits. The desired response features are calculated from the knowledge of the stage-1 coefficients.

The inter-sweep modelling is concerned with the variation *among* the variables or, the relationship in the sweep specific parameter vectors with the remaining engine operating variables. The second stage or *Global Model* is multivariate in nature as contrast to the first stage which deals with the relationship of one variable with the response.

In global model the focus is on the analysis of the changes in the response features among sweeps. The MLP neural models is selected for the each response features. At the second stage we refers to model these curves change as a function of other engine parameters, the global variable *i.e.* (N, L, ICP, ECP). Each test is taken at different point in the global variables, and each response feature is first considered separately - for *'univariate'* modelling. Once the model is obtained using univariate methods, *'multivariate'* maximum likelihood techniques is used to estimate the corresponding parameters. The model fits to the response features are visualized graphically, and considerable attention is paid to the model coefficients to ensure that the model behaves in accordance with the physical theory. This is referred to *'internal validation'*. An internal validation is performed to ensure

that the model behaves accordance to the physical theory(2). Also, the model is validated on a fresh data , that is never been used in the modelling to investigate its predictive performance, in *'external validation'*. Both of these validation studies are important steps in two-stage modelling process.

5.2 Model Structure

A simple block diagram form of two-stage model structure is shown in figure 5.2. At first stage the model represents the variation among measurements within a sweep (*'intra-sweep variation'*) by modelling the relationship between the spark advance and brake torque. At second stage, neural network models are used to describe the systematic variation in the response feature behaviour across sweeps (*'inter-sweep variation'*). These models are use to describe the shape of the spark advance verses brake torque variation with (N, L, ICP, ECP). These different variance components are accommodated within the framework of hierarchical statistical model.

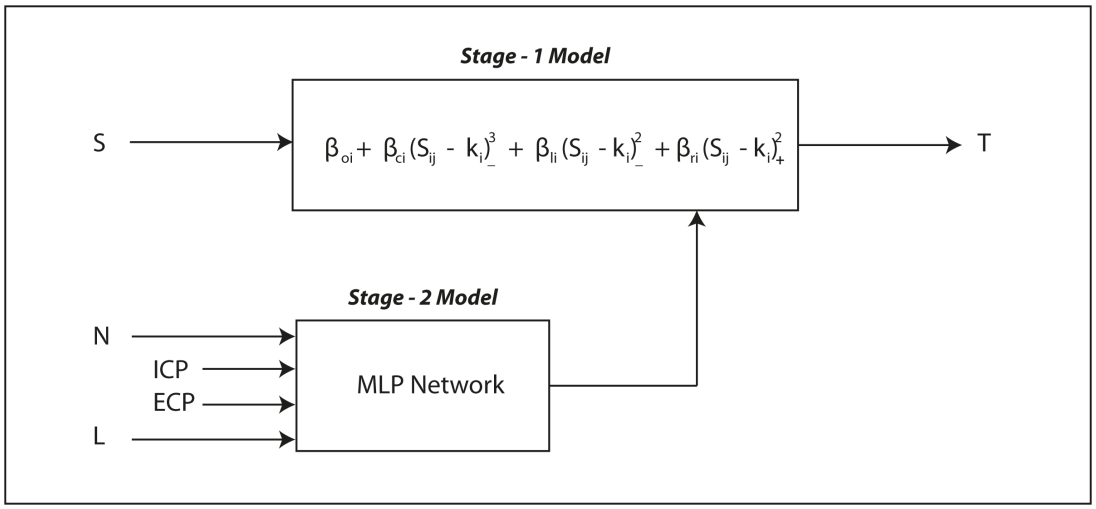


Figure 5.2: Two-stage model structure

The characteristic geometric features of the curves or response features are of the

primary interest, and not the parameters associated with the stage-1 model which has no direct engineering interpretation. Therefore, the variation in the response features across sweeps is modelled rather than the curve fit coefficients themselves. The decomposition of the modelling into two stages simplifies the overall analysis (48; 49). Constructing models to take into account the way the data is collected helps build good models that are easily interpretable and high confident.

The detailed description of the two-stage models for engine can be found in (48; 49; 50; 58), here only a brief description is provided for the purpose of this study.

5.2.1 Two-Stage Model Definition

The first stage concerns the modelling of the output variables over spark sweeps. Let T_{ij} denote the j^{th} measured brake torque value, $j \, \epsilon \, (1, n_i)$, for the i^{th} spark sweep S_{ij}, $i \, \epsilon \, (1, m)$. At the Stage-1 the specification of the i^{th} sweep is

$$T_{ij} = f_i(S_{ij}, \beta_i) + e_{ij} \tag{5.1}$$

where $\beta_i \, \epsilon \, \Re^r$ is the sweep specific parameter vector, and e_{ij} is the within-sweep error variation vector for $j = 1, \ldots, m_i$, *and* m_i is the number of observation in the i^{th} spark sweep, about 11 in this study. It is assumed that $E(e_{ij}|\beta_i) = 0$, with E the usual expectation operator. The Equation 5.1 considers that in stage-1, fit function may vary from sweep to sweep.

For the i^{th} sweep, collection of responses into the vector $T_i = [T_{i1}, \ldots, T_{in_i}]^T$ and its corresponding errors into the vector $e_i = [e_{i1}, \ldots, e_{in_i}]^T$. Similarly, define the vector f_i as $f_i(\beta_i) = [f_i(S_{i1}, \beta_i), \ldots, f_i(S_{in_i}, \beta_i)]$. The data for the i^{th} sweep can be summarized in the compact form as

$$T_i = f_i(\beta_i) + e_i, \quad e_i|\beta_i = \mathbf{N}_{n_i}[0, \sigma_i^2 I_{n_i}] \tag{5.2}$$

here **N** is the normal distribution. The Equation 5.2 shows a relatively very simple sweep covariance model, describe both the systematic and random variation associated with measurements taken during the i^{th} sweep.

In the second stage of the model, the sweep-specific parameters vector, β_i in Equation 5.1 or 5.2 is related to the other engine parameters, (N, L, ICP, ECP), through sweep output features determined in terms of these parameters.

$$\beta_i = \mathbf{d}(\mathbf{a_i}, \theta) + \gamma_i \tag{5.3}$$

where d is a p-dimensional vector valued function, $\theta \in R^r$ is a vector of fixed parameters, a_i is a suitably dimensioned matrix of level-2 covariates. Also, $gamma_i \in R^P$ is a random effects distributed as $\mathbf{N}(0, \Gamma)$, with $\Gamma \in R^{PP}$ being the inter-sweep covariance matrix.

Generally, its a geometric features of the curve that is of interest, rather then the curve fit parameters β_i themselves which have a little interpretative value. These characteristics features are defined as *'response features'*(56). The variation in the response features are modelled across sweeps rather than the β_i for engineering plausibility of their links to these variables. These response features will be related to the fit parameters through a non-linear vector valued function, $p_i(\beta_i)$ say. Thus, the second stage model is relating the systematic variation in $p_i(\beta_i)$ to changes in the remaining engine parameters.

The second stage model assume that the response features p_i can be approximated to a linear parameter statistical model in (N, L, ICP, ECP), but not in Spark S, with additive independent normal variation (or error) γ_i having a common variance matrix Γ_i; thus

$$p_i = a_i\theta + \gamma_i, \quad \gamma_i \sim \mathbf{N}_r(0, \Gamma_i) \tag{5.4}$$

where a_i is a model specific matrix depends on simple functions of N, L, ICP and ECP, the mean values of the engine parameters for the i^{th} spark sweep. Similarly, θ is the stage-2 model parameters vector to be estimated. The form of a_i will depend of set of models used to relate p_i to N, L, ICP and ECP, and involving key engineering inputs. As spark does not feature in the design space for a_i, its data can be collected in highly effective and economical manner involving just few value of each of these variables, which otherwise could complicate matters in different sweeps.

Combining both the stages without considering the estimation, the final determination form of the model is

$$y = f(S; h(N, L, ICP, ECP)) \tag{5.5}$$

while h is a function relating the scalar form of vector for β. This is a relationship between the sweep output variable y and inputs spark S and other engine parameters shown. Both of the error terms of e_i and b_i of stage-1 and stage-2 are included in Equation 5.5, making it more statistical relevant.

Two stage model is mainly used for prediction (49), which mean model can produce sweeps that are good approximation of the real sweeps, for the dataset that is not used in construction and fitting of the model. This could only be achieved if the model is a good approximation to the intrinsically true model.

5.2.2 Local Model

At local model or stage-1, the primary focus of the model is to find a fit function that could accurately describe the silent feature of the spark sweep. Local models are fitted to each test, in different places across the global space. This model is utilized to calculated the response features for the second-stage modelling.

Local models find the best fit of a curve to the data in each test. Each test

in this case is a torque against spark angle sweep, with speed, load, intake and exhaust cam phase at a constant value for each sweep. Figure 5.3 shows a graphical representation of the typical features of the torque vs spark sweep.

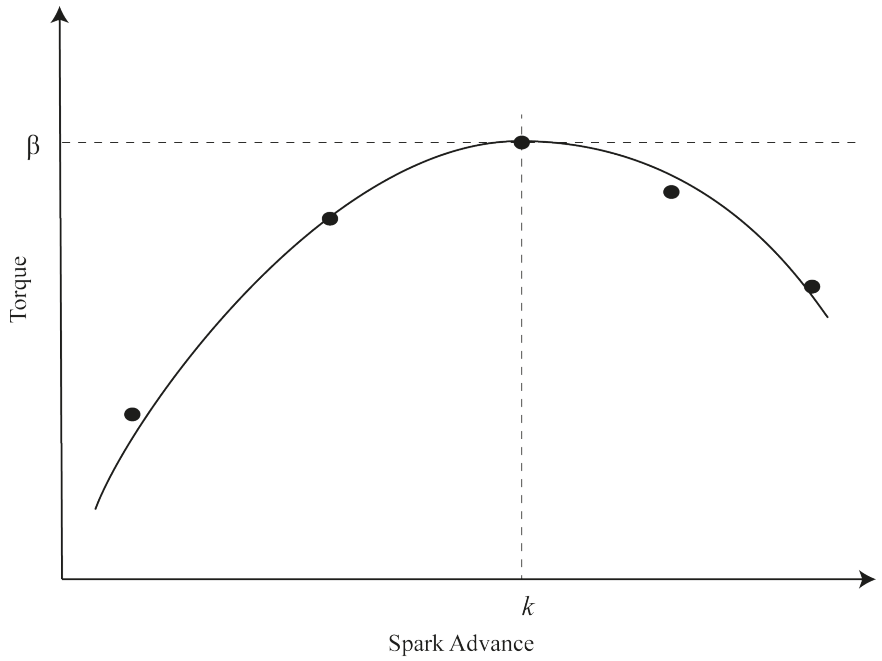


Figure 5.3: Spark advance vs Torque

These local models provide the coefficient to generate global models. The equation describing those local model curve have certain coefficients for Maximum Brake Torque (MBT) and peak torque (PKTQ) The MBT spark is one of the primary model used within the calibration methodologies, with all the features calibrated to operated at this setting. By modelling the torque response, the MBT spark advance could be identified. These coefficients become the data to which the global models are fitted, and give a much better perceptive of a feature such as MBT spark varies through the global factor space than some obscure curve fit parameter.

The shape of spark vs torque is well understood. The torque rises to its maximum as the spark is advances throughout its range, and then falls. Spark knock is not

modelled in simulation engine model used this study, so spark advance was not limited as it would be in dynamometer based testing, where knock-limited spark advance is typically modelled as a separate response and used later in optimisation. Each sweep is inspected for any outliers; and fitting of different alternative model for minimizing the local RMSE statistic.

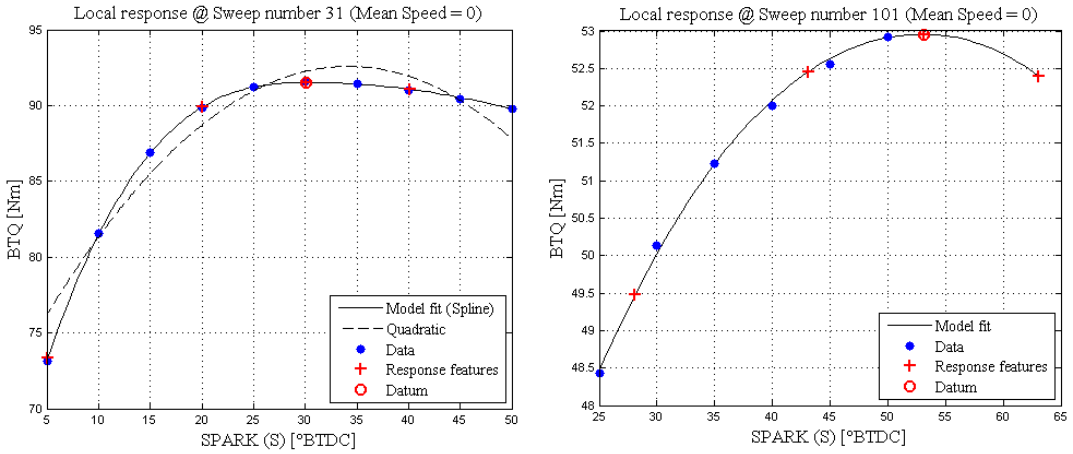
Holliday(49; 50) has suggested using a segmented or spline model for the representation of these fits, where a quadratic curve fit function is inappropriate and exhibit significant bias. A segmented polynomial or spline curves are useful for fitting these shapes, where different curvature is required above and below the maximum. Holliday suggested stage-1 model has the following form:

$$E(T_{ij}^{b}) = \beta_{0i} + \begin{cases} \beta_{li}(s_{ij} - k_i)^2 & \text{if } s_{ij} \leq k_i \\ \beta_{ri}(s_{ij} - k_i)^2 & \text{if } s_{ij} \geq k_i \end{cases} \tag{5.6}$$

However, Cary(58) has added a cubic term to compensate the bias exhibit in Equation 5.6 due to a high levels of spark retard for some sweeps. This results in the equation of the form:

$$E(T_{ij}^{b}) = \beta_{0i} + \begin{cases} \beta_{ci}(s_{ij} - k_i)^3 & \text{if } s_{ij} \leq k_i \\ \beta_{li}(s_{ij} - k_i)^2 & \text{if } s_{ij} \leq k_i \\ \beta_{ri}(s_{ij} - k_i)^2 & \text{if } s_{ij} \geq k_i \end{cases} \tag{5.7}$$

The Equation 5.7 is a local regression model, that relate brake torque to spark advance with all other parameters (N, L, ICP, ECP) held constant. Figure 5.4(a) shows a local model at engine speed of 2685 RPM, where maximum break torque is shown in a circle while values of torque at spark advances and retard are represented by a cross. However, it is not a full presentation of spark sweep, and several sweeps were terminated prior to achieving maximum brake torque, or just after, due to the onset of moderate or heavy detonation. Similarly, some of the sweeps exhibits only limited information to the left of maximum where exhaust gas temperature or combustion stability constraints apply. In this case Equation 5.7 is inappropriate.



(a) Local model fit (Spline)

(b) Local model fit (Quadratic)

Figure 5.4: Spark sweeps with fitted curves to data from (3) local model

However, the Equation 5.7 do not restrict the sweep specific response features calculation, and at local model desired response features can be calculated from the coefficients associated with the alternate model. For example, one of the response feature of interest is the spark advance yielding maximum brake torque, denoted by MBT. It is sometime possible that the $MBT_i (= k_i)$ cannot be achieved for some spark sweeps as it will be outside the range of possible spark advances (shown in figure 5.4(b)) due to limitation by the onset of detonation.

In this case, the Equation 5.7 cannot be used to fit the data, but rather a quadratic relationship $\beta_{oi} + \beta_{Qi} s_{ij} + \beta_{Li} s_{ij}^2$ can be used to fit the data for the i^{th} sweep, which provide a good approximation to the data in this case. MBT_i is calculated now as $MBT_i = -\beta_{Li}/2\beta_{Qi}$, which may represent substantial extrapolation of the data. The MBC toolbox automatically calculates and applies the appropriate covariance matrix Σ_i dependent on the choice of $f_i(\beta_i)$.

5.2.3 Response Feature Selection

It is rather the characteristic geometric features of the curve that are of interest then the curve fit parameters which usually do not have any intuitive interpre-

tation from an engineering perspective. The terminology *"response features"* of Crowder and Hand (56) is used to describe these geometric features of interest. In general, the response features will be related to the fit parameters through a non-linear vector valued function, $p_i(\beta_i)$.

Once the $f_i(\beta_i)$ for separate sweep profiles and the corresponding $p_i(\beta_i)$ for response features is defined, the focus tends toward the selection of a single model for predictive purpose. At stage-2, the response feature vector, $\hat{p}(\beta_i)$ is evaluated at different value of global variables. Then the corresponding $\hat{\beta}_i$ is calculating by inversing the function $\hat{p}(\beta_i)$, to determine the appropriate response features. Figure 5.5 shows the relationship between the global models and global factors.

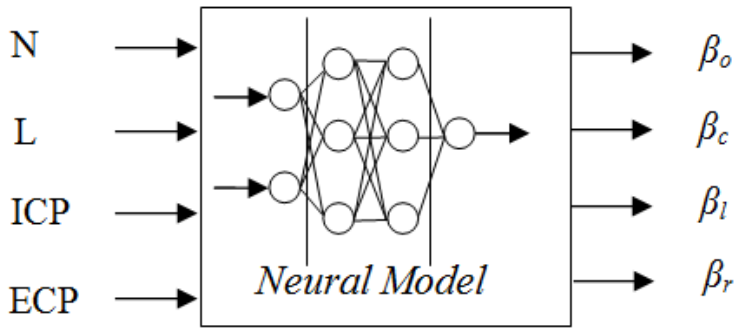


Figure 5.5: Global model form

In the second stage of the model, the variation of the torque response with respect to spark sweeps is modeled. A typical curve of the torque increases to the maximum and then decrease again, except for few sweeps where the relationship is more monotonic than concave.

The curve coefficients k_i and β_{oi} in Equation 5.6 and 5.7 having a direct engineering interpretation can thus be considered as response features. k_i is the spark advance which gives the maximum brake torque (MBT_i) and β_{oi} the corresponding torque produced at MBT_i (denoted by PKTQ_i). The influence of spark timing to retarding or advancing from MBT_i is related to three additional

response features, ΔLESS10$_i$, ΔLESS25$_i$ and ΔPLUS10$_i$. These are observed torque reduction from the maximum when the spark timing is retarded 10 and 25 degree from MBT$_i$ and advanced 10 degree from MBT$_i$, respectively.

The spark sweeps is therefore characterize according to five transformations, i.e., MBT, PKTQ, Δ LESS10, ΔLESS25 and ΔPLUS10. The ΔLESS10$_i$ and ΔPLUS10$_i$ measure the asymmetry of the response profile about the maximum, while ΔLESS25$_i$ measures how rapidly the response falls away under heavy spark retard.

The corresponding response feature vector for the i^{th} sweep using this nomenclature, is written as:

$$p_i = \begin{bmatrix} MBT_i \\ PKTQ_i \\ \Delta LESS10_i \\ \Delta LESS25_i \\ \Delta PLUS10_i \end{bmatrix} = \begin{bmatrix} k_i \\ \beta_{0i} \\ 10^2\beta_{LQ_i} - 10^3\beta_{LC_i} \\ 25^2\beta_{LQ_i} - 25^3\beta_{LC_i} \\ 10^2\beta_{RQ_i} \end{bmatrix} \tag{5.8}$$

The above Equation 5.8 is easily inverted to allow Equation 5.7 to be written in terms of g_i as follows

$$E(T^b_{ij}) = PKTQ_i + \begin{cases} \left[\dfrac{\Delta LESS25_i}{9375} - \dfrac{\Delta LESS10_i}{1500}\right](s_{ij} - MBT_i)^3 & \text{if } s_{ij} \leq MBT_i \\ \left[2\left(\dfrac{\Delta LESS25_i}{1875}\right) - \dfrac{\Delta LESS10_i}{60}\right](s_{ij} - MBT_i)^2 & \text{if } s_{ij} \leq MBT_i \\ \Delta PLUS10_i\,(s_{ij} - MBT_i)^2 & \text{if } s_{ij} \geq MBT_i \end{cases} \tag{5.9}$$

5.2.4 Global Model

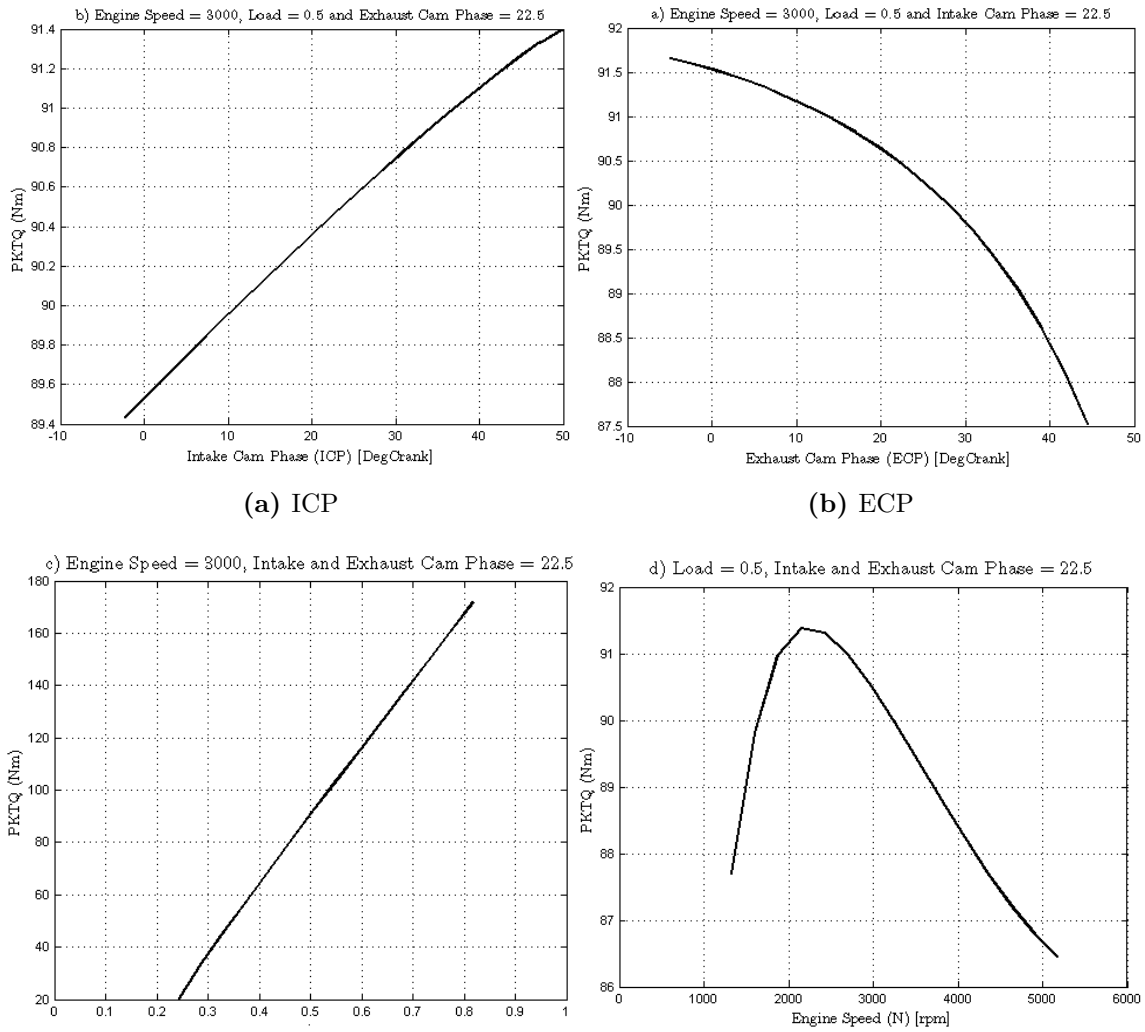
Global models are the best fit of a curve to the values of local model coefficients for each test. This is repeated for each coefficient, producing several global models fitted to different coefficients of the local models. These coefficient are the response features of the local models. The response features are modelled

in stage-2 as a function of the covariates; i.e., N, L, ICP and ECP, for each of the torque vs spark advance local model regression using MLP neural network. The systematic variation in the response features are summarized in the second stage using neural network. A neural network was chosen because of its ability to accurately model fine variations in torque due to tuning effects, without the over-fitting problems associated with other approaches such as high-order polynomials. The multi-layer perceptron network is fitted with Bayesian regularization method (i.e. *trainbr*). The *trainbr* is a network training function that updates the weight and bias values according to Levenberg-Marquardt optimization (97). It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes (122). The architecture of the MLP network contain one hidden layer, with 10 number of neurons in the layer. However, different network can be used to train different response features data, and the architecture of the network can be changed accordingly to obtained best fit of the model with minimising the fit error.

A comparative study was also carried out with an hybrid RBF model fitted in the global model, that is used in reference (3; 58). The radial basis function regressions is fitted using Multi-Quadratic technique. The model algorithm is initialised with 50 RBF centers in order to achieve a good fit and at the same time avoid over-fitting.

$$\begin{bmatrix} MBT_i \\ PKTQ_i \\ \Delta LESS10_i \\ \Delta LESS25_i \\ \Delta PLUS10_i \end{bmatrix} = \begin{bmatrix} a_{MBT_i} & 0 & 0 & 0 & 0 \\ 0 & a_{PKTQ_i} & 0 & 0 & 0 \\ 0 & 0 & a_{LESS10_i} & 0 & 0 \\ 0 & 0 & 0 & a_{LESS25_i} & 0 \\ 0 & 0 & 0 & 0 & a_{PLUS10_i} \end{bmatrix} \begin{bmatrix} \theta_{MBT_i} \\ \theta_{PKTQ_i} \\ \theta_{\Delta LESS10_i} \\ \theta_{\Delta LESS25_i} \\ \theta_{\Delta PLUS10_i} \end{bmatrix} \tag{5.10}$$

Figure 5.6 shows the response feature PKTQ plotted against the four covariates. It shows a very strong linear relationship between PKTQ with ICP and Load,



(a) ICP **(b)** ECP

(c) Load **(d)** Speed

Figure 5.6: PKTQ response feature trend analysis

as it would be expected. If there is more air going into the engine there should be more power coming out. PKTQ has a cleared curved relationship with engine speed, with a maximum torque obtained at a peak around the middle of the speed range at 2200 rpm. The engines of this type are designed to operate at their best near the middle of their speed range. Both intake and exhaust cam phase have got an increasing and decreasing trend respectively, with fixed load and speed.

A residual plot is a plot of the standardized residuals against the levels of another variable, the choice of which depends on the assumption being checked. Plots like this are useful for evaluating the assumption of constant error variance as well as the adequacy of the model. In general, lack of fit is indicated if the residuals exhibit a non-random pattern about zero in any such plot, being too often positive for some levels of the independent variable and too often negative for others. For ΔLESS10 plot revealed observation 30 revealed to be an obvious outlier, shown in Figure 5.7 and no obvious explanation could be determined from the local model analysis. The presence of this outlier appreciably degraded the fit and quality of the residual diagnostic plots. Therefore, this observation was removed.

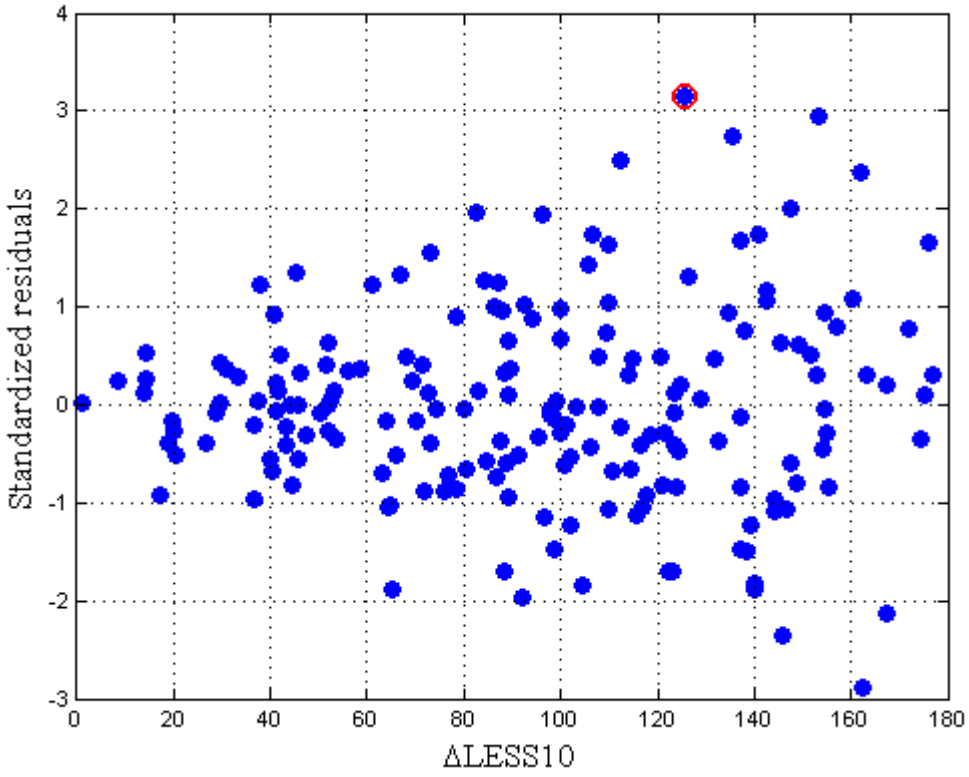


Figure 5.7: Residual diagnostic plots

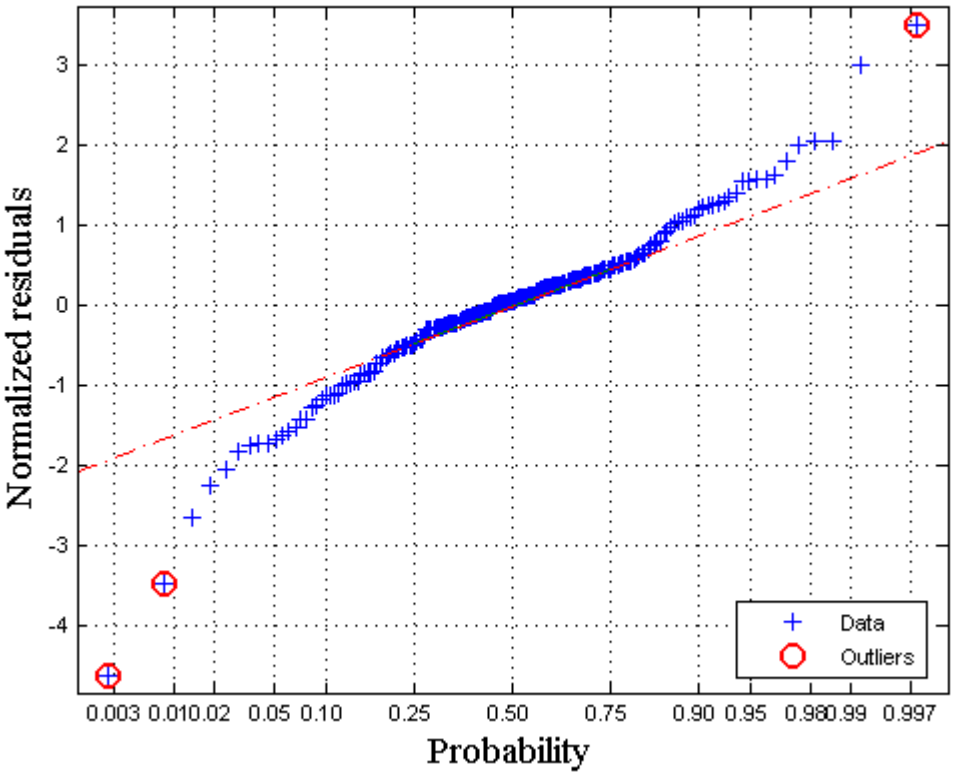


Figure 5.8: Residual diagnostic plots for $\Delta LESS25$

For ΔLESS25 fit, three observations, 16, 108 and 110 shows a considerably high error from the model fit. Figure 5.8 show residual diagnostic plots for ΔLESS25 Improvement were observed after setting aside these observations. Also, in the same pattern ΔPLUS10 plots were analyse for its behaviour.

A separate neural model is fitted to each response features. Table 5.1 shows the relevant summary statistics for each of the 5 regressions using MLP and RBF models.

Typically, a model consists of one or more equations. The quantities appearing in the equations are classified as variables and parameters. The distinction between these is not always clear cut, and it frequently depends on the context in which the variables appear. A parameter is a quantity that serves to relate functions

Table 5.1: Models fit summary statistics

Response Feature	Model Type	Observation	Parameters	RMSE	BIC	AIC_c
MBT	Radial Basis Function	186	56	1.911	457.713	329.868
	Multi-Layer Perceptron Network	183	111	1.361	525.668	527.806
PKTQ	Radial Basis Function	186	52	0.501	-43.786	-170.514
	Multi-Layer Perceptron Network	183	111	0.522	174.226	154.356
PLUSS10	Radial Basis Function	186	60	0.760	5.658	-129.556
	Multi-Layer Perceptron Network	187	111	0.414	87.677	67.847
LESS10	Radial Basis Function	186	59	0.969	223.104	89.933
	Multi-Layer Perceptron Network	187	111	0.961	402.656	382.865
LESS25	Radial Basis Function	186	60	4.797	827.847	692.434
	Multi-Layer Perceptron Network	187	111	5.151	1030.591	1010.761

and variables using a common variable when such a relationship would be difficult to explicate with an equation.

Usually a model is designed to explain the relationships that exist among quantities which can be measured independently in an experiment; these are the variables of the model. To formulate these relationships, however, one frequently introduces some unknown constants or coefficients, which stand for inherent properties of nature (or of the materials and equipment used in a given experiment), these are the parameters (123). These parameters often have a physical interpretations, a major aim of the investigation is to estimate the parameters as precisely as possible, and a further aim is to test the fit of the data to the model.

If several models fit the data equally well, it the simplest model that is chosen. For example, in prediction, or in finding the maximum and minimum values of the curve or the slope at particular points, than splines (segmented polynomials) may be appropriate (123). However, with the model having complex structure, these simplest polynomials models is having very little use, and the neural network models does a great job in estimating the unknown parameters.

Table 5.1 shows comparison of the two different models use. The number of parameter in case of MLP based neural model is double as compared to RBF model, which means MLP models require a smaller number of parameters then the RBF network for the same degree of accuracy and as the number of parameters increases the accuracy of the model increases. The root mean square error (RMSE) for the MBT for MLP is 1.361° compared to that of RBF which is 1.911°.

5.2.5 Exhaust Temperature and Residual Fraction Model

The MBT spark was the primary feature within the DOE, it is therefore beneficial to provide an example of some secondary feature to show the quality of the response models generated. The exhaust temperature model is an example of such a feature. A two-stage modelling approach was used with a cubic fitted to the local response, here the exhaust temperature was modelled as a function

of spark advance. Table 5.2 reveals that the local RSME was less than 4 Deg F, however, it should be recognised that the use of a cubic response within the local spark range proved satisfactory, if the model response was inspected outside this range the model would witness a significant climb towards infinity in some cases. The temperature was measured at the point where all four manifold runners met.

Also, in addition to the MBT and exhaust temperature model, the internal residual fraction of burned gas at intake valve was modelled as a function of the global variables N, L, ICP and ECP to act as an indicator of combustion stability. Here also, two type of model were fitted, an RBF and a NN model, to the calculated internal residual corresponding to the maximum torque points in the torque/spark sweeps of the the survey. Table 5.2 shows the residual differences between predicted and measured residual fraction, which is within 1% residual.

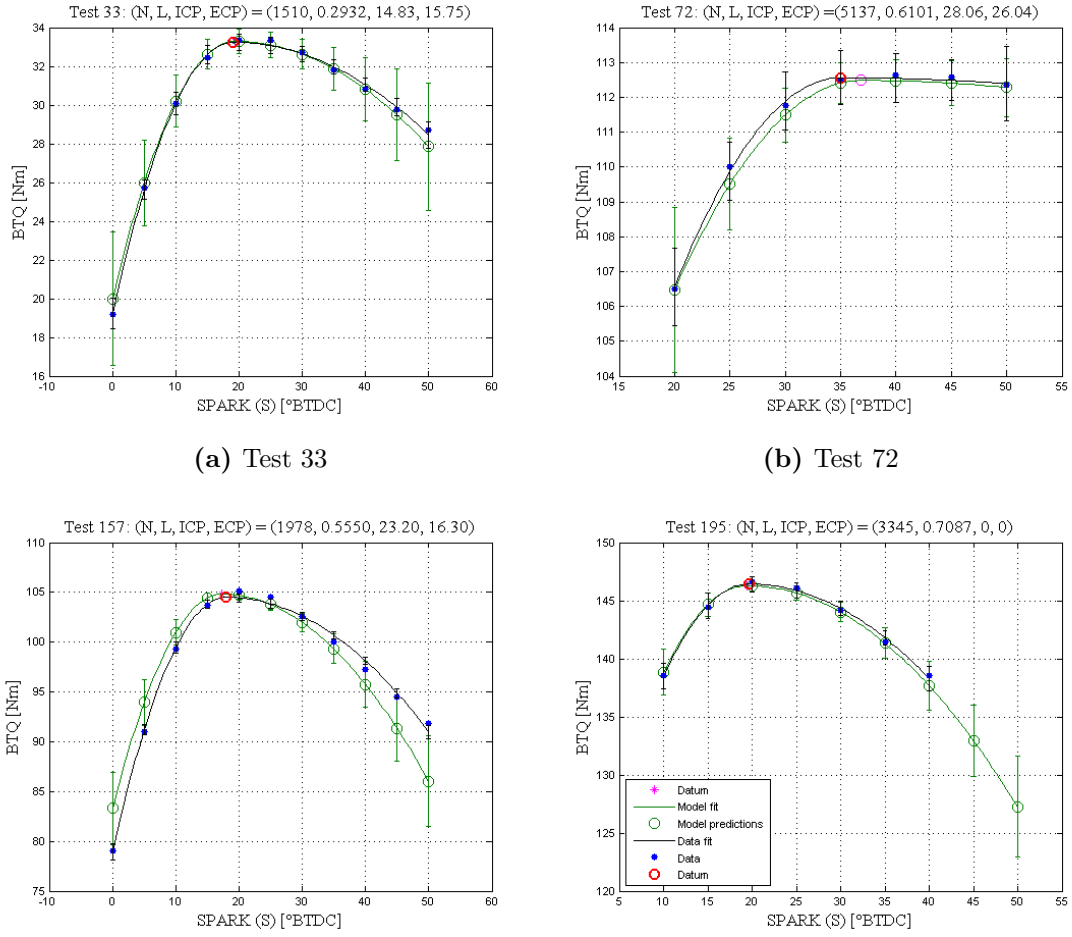
Table 5.2: Univariate regression summary statistics

Response Model	Model Type	Local RMSE	Two-Stage RMSE	Two-Stage T^2	Validation RMSE
BTQ	RBF	0.358	1.146	3.214	4.681
	MLP	0.358	2.033	3.126	4.546
EXTEMP	RBF	3.642	7.542	2.299	33.007
	MLP	3.642	7.235	0.759	49.487
RFRAC	RBF	0.03	0.231	2.792	2.484
	MLP	0.038	0.293	2.817	2.283

5.2.6 Model Fitting Summary

Table 5.2 compare two different modelling techniques. Comparison of the two models rely upon several metrics, these includes inspection of residuals, root mean square error (RMSE), response surface and the inspection of the model on

the validation data. The data not used in the model development is used as a validation run against the model prediction. And equivalent model based on Radial basis function (RBF) method used by Cary (58) is also constructed to facilitate the comparison with the current model based on neural network. Inspection of Table 5.2 shows that there is little to separate the statistics from an engineering viewpoint, with both models show remarkably small errors. The neural network based models, however exhibits a smaller RMSE. However, there is no intent to imply here that the neural networks generally out perform the RBF models.



(a) Test 33

(b) Test 72

(c) Test 157

(d) Test 195

Figure 5.9: Training data fit to various test sweeps

Finally, the two-stage model is examined by comparing it with the local fit and with the data. The local torque/spark curve at an operating point is reconstructed by taking the values of MBT and peak torque and the curvature from the two-stage model, and then validated this reconstructed curve against the original fit and the data. The two-stage model can also predict responses between tests, for new sweeps at intermediate values for which there is no data.

Figure 5.9 presents some example fits for the model to a selection of sweeps from the training data. These plots are considered representative for the entire set of 189 sweeps. The two-stage model shows an accurate fit when compared to the local sweeps. This is a good sign that the engine behaviour is well described by the model across the global variables.

The individual torque predictions is calculated with Equation 5.7, assuming that the response features are independent on the global variables (i.e., N, L, ICP, ECP). This assumption is based on the fact that the point has been calculated on the actual observed values of the global variables rather than the averaged for the sweep. This result is in lack of smoothness in the predictions. Also, the comparison between the models should be made objectively, with some compromises should have been taken for good reason incurring potential error.

5.3 Developed Model Validation

The model is validated to a fresh data, to excess the validity and predictive performance of the model prior to its use. This process is referred to as *external validation* (58).

The process involves the collection of data for the study of model's predictive performance, i.e., the model predicts the response characteristic at input configuration that has not been used for training. The validation design should be compact, but should exercise over the full range of conditions associated with its intended use.

Therefore, to access the accuracy of the two-stage model developed, a validation design is also constructed on the space-filling design method with number of arbitrary engine operation conditions. The design contain 25 points for the same four input factors for each of the spark sweep.

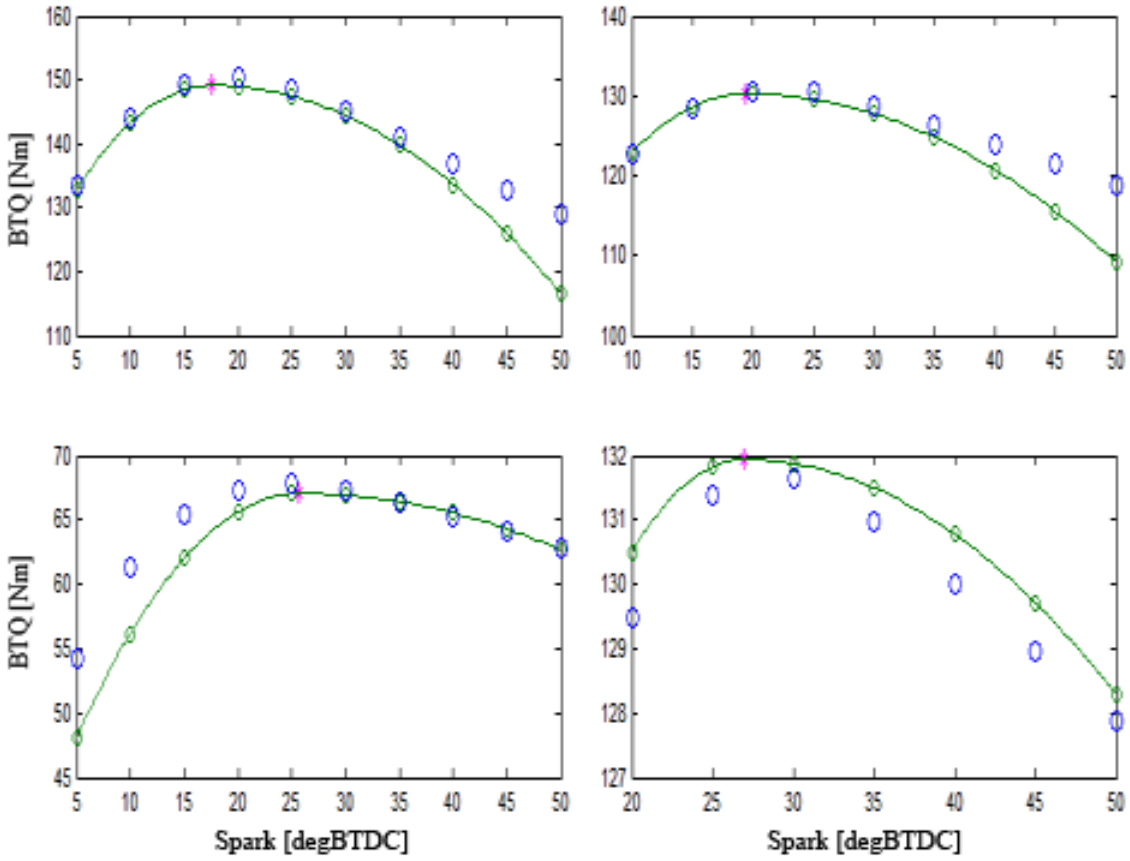


Figure 5.10: External validation

Figure 5.10 shows the external validation of model developed in Section 5.1. These plots are representation of the validation sweeps created in the validation design. Again, here the local torque/spark curve at an operating point is reconstructed by taking the values of MBT and peak torque and the curvature from the two-stage model, and then validated this reconstructed curve against the original fit and the data. A comparison between the torque model and the validation points

shows a good correlation. The root mean square error for these data is 4.546 [Nm] for brake torque.

5.4 Summary

This chapter presented a steady state engine mapping case study using two-stage regression approach for the analysis of engine brake torque data collected on a 2.2L inline 4 cylinder, naturally aspirated Dual Overhead Cams (DOHC) and subjected to the different constraints, to ensure that the engine runs in its region of operatibility. The primary focus was to accurately model the brake torque as a function of spark timing at level-1 and the response features as a function of engine speed, normalised induced air charge (or *load*), the intake and exhaust cam phase. Additional responses of exhaust port temperature and residual fraction is also modelled in the same manner.

At level-1, a systematic behaviours in the brake torque with spark advance was modelled by deriving a sweep-specific relationship. A segmented polynomial curve is fitted, with a specification of model at level-1 as:

$$E(T_{ij}^{b}) = \beta_{0i} + \begin{cases} \beta_{ci}(s_{ij} - k_i)^3 & \text{if } s_{ij} \leq k_i \\ \beta_{li}(s_{ij} - k_i)^2 & \text{if } s_{ij} \leq k_i \\ \beta_{ri}(s_{ij} - k_i)^2 & \text{if } s_{ij} \geq k_i \end{cases}$$

At level-2, the focus is on representation of the variation in the response features among sweeps. The neural network models based on Multi-Layer Perceptron (MLP) has been introduced to adequately describe the response feature variations at the second stage. The resulting model affords an excellent fit to the data with accurately representing the complexity of the MBT response feature with engine speed.

A comparable model based on Radial Basis Function (RBF) was also developed along the MLP model. It has been shown that, the MLP model has a slight improvement prediction capability than the RBF models, with number of parameter

increases in case of MLP for the same number of observations. Hence, for the approximation of a nonlinear input-output mapping, the MLP based neural network require a smaller number of parameter then the RBF network for the same degree of accuracy. A model accurately predicting the desired response features over the entire region of operability can be developed using this approach.

6

Neural Network Based Transient Engine Model

In previous chapters a steady state engine mapping case study illustrating the application of two stage regression techniques, and their associated design of experiments was discussed. Models for engine brake torque, exhaust temperature and residual fraction were developed and validated. Also, at stage-2 each of the response features were modelled using MLP networks. The MLP based engine model show great improvement in the prediction capability of the steady state two-stage regression approach. And therefore, the use of MLP network make it possible to perform the calibration studies over a large range of operating domain, with good model fit and prediction.

However, the steady state engine mapping or calibration optimisation are not suited to the situation in which the prevailing emissions standard is transient standard, such as EPA Heavy Duty Transient Cycle (HDTTC), EPA Smoke Test, EURO III – Load Response Test and the FTP – 75 test for light-duty engine. In these case, the legislative requirement is based on the engine performance over a dynamic cycle, which implies that the engine calibration optimisation must be performed using transient or dynamic engine model (30).

Thus, this chapter presents some original work regarding the development of transient engine model for engine calibration, based on two-stage regression approach. The two-stage regression model discussed in previous chapter for the steady state condition is modified with an introduction of an additional dimension of time in its hierarchy. The modification allows for the presence of identification signal at stage-1 in the model, and hence, for the transient engine calibration application.

6.1 Transient Modelling

Most of the model based calibration effort has traditionally been focused on steady state operation. And only a few researchers like (30; 42; 44; 45; 46) have ventured to address transient calibration. Atkinson et al. (30; 42; 44) used neural networks to predict transient engine operation, by using a hybrid equation-based and neural network-based data-driven technique to produce an engine model for calibration and optimisation. Their work include the utilisation of DoE for the data collection and using it for the development of engine model for engine calibration. However, the design of transient engine experiment for the model results in data point that is still considered too expensive to run. Brahma et al. (45) have used empirical modelling for transient emissions and response for optimisation. Their work was limited to only using full quadratic global regression for modelling and prediction, which result in poor performance at some response (Smoke and PM).

Transient modelling is different from the steady state modelling in various ways. The most important is the data acquisition and processing in transient condition is highly complex and is very important in view of model development. Also, the model developed for the steady state condition might not be suitable for the transient data.

6.1.1 Transient Two-Stage Regression Model

The two-stage model discussed in Section 3.3 assumed that the covariate vector a_i summarizing individual characteristics is constant across the observations on

individual i, and which further specifies that the value of the regression parameter β_i for individual i remains fixed for that individual over the course of observation. In some cases, particularly in engine transient phenomena such as engine warm up and fuel dynamic response characteristics, individual specific information may change during the course of observations, to exhibit corresponding changes at different time.

A slight modification to the general structure at stage-1 in the hierarchy is required to permit the individual regression parameters, to depend on changing individual-specific information while handling time-varying individual attributes (55; 123). Let a_{ij} represent the vector of covariate values for individual i corresponding to the j^{th} condition of measurement x_{ij}, and let β_{ij} be the value of regression parameter for individual i at conditions j. Hence, the hierarchical model for the presence of *time-dependent covariates* can be written as follow (55):

Stage 1 (Intra-individual variation):

$$y_{ij} = f(x_{ij}, \beta_{ij}) + e_{ij}, \quad Var(e_{ij}) = \sigma^2 g^2(f(x_{ij}, \beta_{ij}), \zeta) \tag{6.1}$$

Stage 2 (Inter-individual variation):

$$\hat{\beta_{ij}} = d(a_{ij}, \theta, b_i), \quad b_i \sim N(0, \Gamma) \tag{6.2}$$

The vector of covariates x_{ij} defined in the stage-1 modelling can be defined by the equation, $x_{ij} = [t_{ij}, u_{ij}]$, where t_{ij} is the time, and u_{ij} is now explicitly included in the formal definition of f_i. The above model can be written as follows:

Stage 1:

$$y_{ij} = f_i(t_{ij}, u_{ij}, \beta_{ij}) + e_{ij}, \quad Var(e_{ij}) = \sigma^2 g^2(f_i(t_{ij}, u_{ij}, \beta_i), \zeta) \tag{6.3}$$

Stage 2:

$$\hat{\beta_{ij}} = d(a_{ij}, \theta, b_i), \quad b_i \sim N(0, \Gamma) \tag{6.4}$$

These covariates values should be available at the same times as the observations, indexed by j within subjects.

6.2 Air Flow Modelling

The engine output torque is controlled by the throttle body in restricting the intake airflow. Throttle plate geometry and parameters are illustrated in Figure 6.1. This type of throttle plate creates a three-dimensional flow field. The throttle plate shaft is usually of sufficient size to affect the throttle open area. The plate is usually completely closed at some non-zero angle ($5, 10,$ or $15°$), to prevent binding in the throttle bore. This is also necessary to provide the desired flow setting during engine idle.

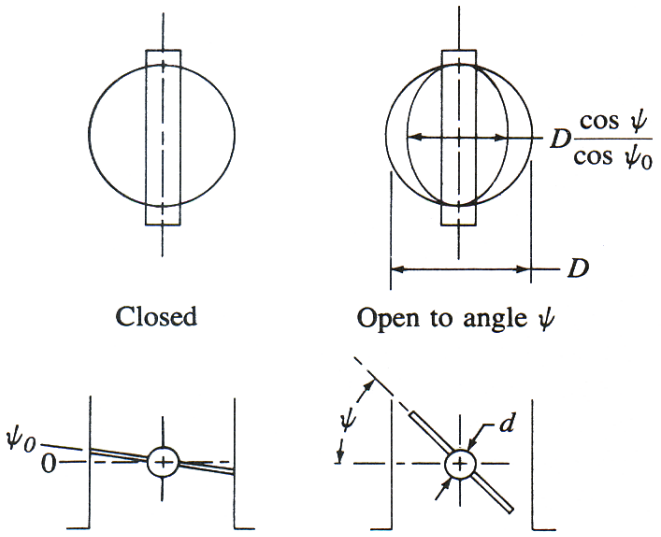


Figure 6.1: Throttle plate geometry (4)

The air flow out of the manifold occurs in a series of sinusoidal pulses, one going to each cylinder. For four cylinder engines, these flow pulses sequence such that the outflow is essentially zero between pulses. When the engine is throttled,

back-flow from the cylinder into the intake manifold occurs during the early part of the intake process until the cylinder pressure falls below the manifold pressure. Back-flow can also occur early in the compression stroke before the inlet valve closes, due to rising cylinder pressure. The flow at the throttle will fluctuate as a consequence of the pulsed flow out of the manifold into the cylinders. At high intake vacuum, the flow will be continuously inward at the throttle and flow pulsations will be small. When the outflow to the cylinder which is undergoing its intake stroke is greater than the flow through the throttle, the cylinder will draw mixture from the rest of the intake manifold. During the portion of the intake stroke when the flow into the cylinder is lower than the flow through the throttle, mixture will flow back into the rest of the manifold. At wide-open throttle when the flow restriction at the throttle is a minimum, flow pulsations at the throttle location will be much more pronounced. The hot wire air flow meter senses the magnitude of the airflow but not the direction, and this leads to gross instantaneous errors in the measured air flow into the engine and therefore in the metered fuel flow.

Similarly, the mass of air in the induction system volume takes a finite time to adjust to the new engine operating conditions when engine load is changed by opening or closing the throttle (4). The air flow to the manifold increases as the throttle open area increases. However, the pressure level in the manifold increases more slowly than would be the case if steady-state conditions prevailed at each throttle position due to the finite volume of the manifold. Thus, the pressure difference across the throttle is larger than it would be under steady flow conditions and the throttle air flow overshoots its steady-state value. The air flow into each cylinder depends on the pressure in the manifold, so this lags the throttle air flow. The transient air-flow phenomenon affects fuel metering. For throttle-body injection or a carburettor, fuel flow is related to throttle air flow. However, for port fuel injection, fuel flow should be related to cylinder air flow. Actual results for the air flow rate and manifold pressure in response to an opening of the throttle are shown in Figure 6.2. The overshoot in throttle air flow and lag in manifold pressure as the throttle angle is increased are evident. Opposite effects will occur for a decrease in throttle angle.

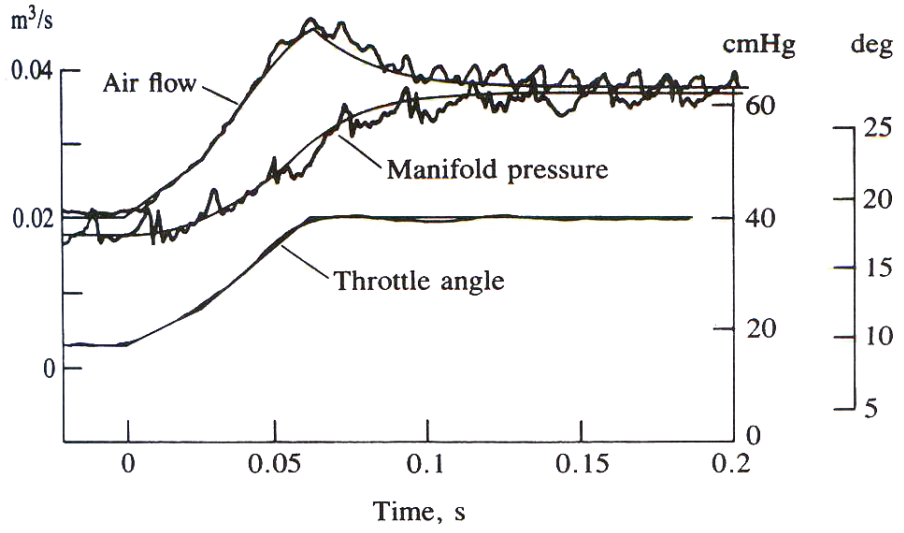


Figure 6.2: Throttle angle, intake manifold pressure and air flow rate past the throttle versus time for 10 deg part-load throttle opening (4)

A failure mode look-up table indexed by throttle angle (θ) and engine speed (N) in the control strategy determined the amount of air flow through the throttle body in a conditions where back-flow is indicated or in areas of high pulsation. However, the table contains the throttle body air flow at a barometric pressure of 29.92 [inHg] and ambient temperature of 100 [°F] at a steady stage condition. The table ignore the transient air flow phenomenon and therefore affects fuel metering.

6.2.1 Experimental Setup

The intention of the experimental work is to initially identify the parameters in the model and then follow this with a series of simple response tests to validate the identified model. In particular the experimental work consist of the transient engine testing according the the experimental design described below and then

using it for the development of the transient engine model using two-stage regression technique. The parameter in the first stage of the model, i.e., the throttle position sweep from its minimum value to the maximum while holding the speed, which is the covariate of the second stage constant.

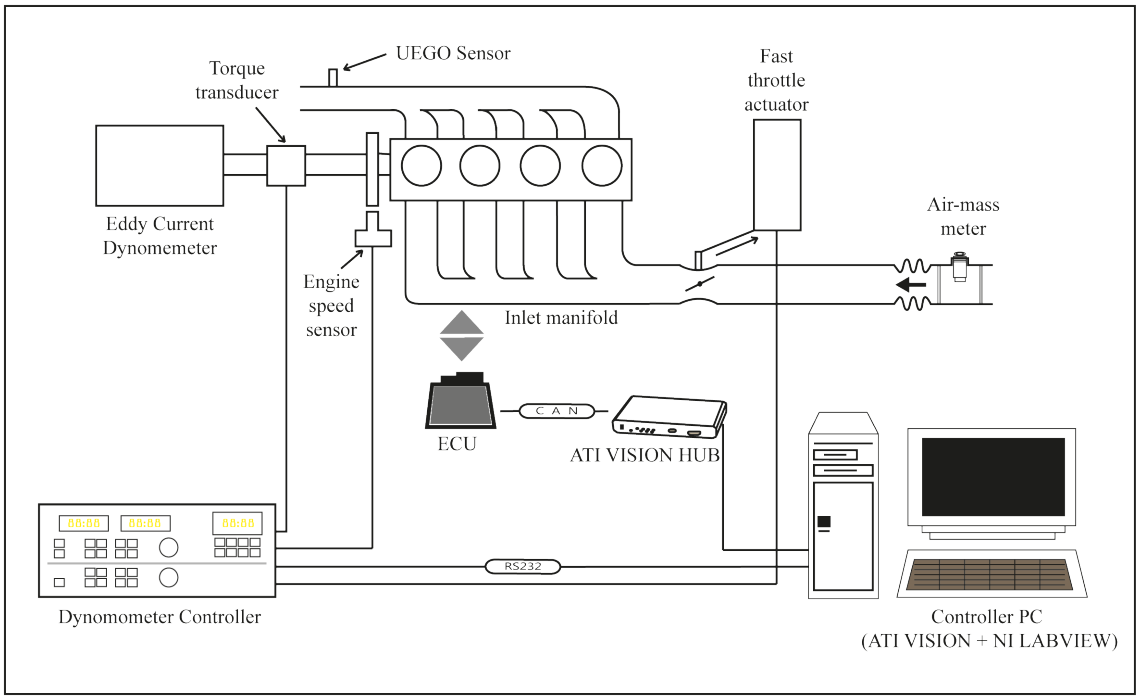


Figure 6.3: Experimental Setup

Figure 6.3 shows the experimental setup. The experimental facilities required involved standard engine test bed equipment used for steady state mapping work suitably enhanced for the transient testing. For this purpose, the PID dynamometer controller is properly tuned for the transient operation before it could be used for the response data collection.

6.2.1.1 Ford Fiesta 1.4 Duratec

The experimental work was carried out on a Ford Fiesta Duratec 1.4 SI engine, fitted with Siemens Electronic Control Unit (ECU), an engine management system prototyping facility necessary to measure, model and control. The engine

was mounted on a test bed and connected via a propeller shaft to a transient dynamometer. Engine control was achieved through an CAN (Control Area Network) system using ATI VISION software to communicate with the ECU. Table 6.1 gives the engine specification,

Table 6.1: Ford Fiesta Engine Specification

Engine Technical Feature:	1.4 liter; 4 cylinder inline; DOHC; 16 valve; alloy cylinder head and block; electronic multipoint fuel injection; electronic throttle
Maximum Power	59 kW (80 PS) at 5700 min^{-1} (rpm)
Maximum Torque	124 Nm at 3500 min^{-1} (rpm)

The engine is mounted on a frame, which hold all necessary connections for engine coolant circuit, fuel circuit, throttle valve positioner, sensor connectors, ECU etc. The throttle valve of the engine is positioned with a servo-actuator that opens the throttle valve to a degree which is proportional to the throttle valve set-point, which is wired as an input to the actuator.

6.2.1.2 Measurement and Control Software

This engine is remotely controlled by the controlling PC via ECU through ATI VISION. This software allows access to Electronic Control Unit (ECU) for calibration, logging measurement data from multiple control sources, analysing collected data and managing calibration data changes. An interface from VISION to the ECU is established via a physical connection with the CAN (Control Area Network) Interface Tools. This interface allows the information contained on the CAN bus to be presented within the VISION software windows. The software supports multiple recorders triggering and storing data simultaneously at user definable channel-by-channel sample rates.

The VISION strategy file stores the device description information, memory images, and system settings for a device. This device is usually the control module under development. The base calibration file is setup with the strategy file, which contain data item values. This calibration file also maintains information on each data item, such as when it was changed and by whom.

Apart from the main controller software for the ECU, a separate interface is developed in NI LabVIEW. This allow to control the software in the transient mode by uploading the experimental design table via an RS232 serial bus to the dynamometer controller. This design table contain information regarding different parameters to be varied at different time settings. The software can also record the dynamometer data, which can be used to match with the data recorded by VISION.

6.2.2 Data Collection Methodology

The experimental flow chart is represented in Figure 6.4. The process of transient data collection start with the design for the transient experiment. The design here again based on space-filling principles was selected, with speed as the only variable at which to experiment. For the design size m (= 23), $m-$unique levels of speed is selected which is distributed uniformly over the range of 900 – 6000 [RPM].

Due to high air induction, five levels of engine speed have been assigned to the range 3000 – 3500 [RPM]. Also, five level of engine speeds were assigned from 950 – 2000 [RPM], due to the effect of back-flow at lower speeds. This leaves with a 13 remaining levels of speed to cover the rest of extensive operating range. The data for this work was acquired from a SI engine equipped with sophisticated software and hardware.

The data was collected at a frequency of 5Hz, which results in 20 measured data at 200ms for each throttle sweep. The transient designed table is fed through the

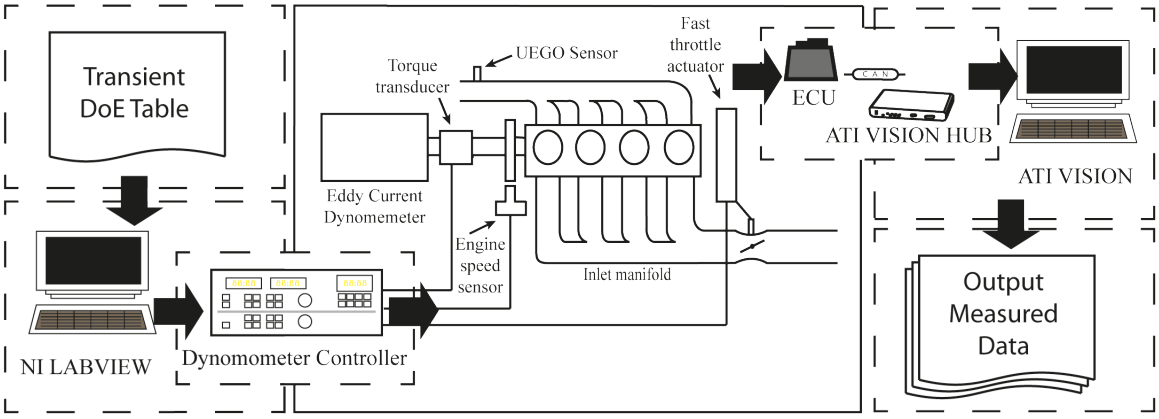


Figure 6.4: Experimental flow chart

NI LabVIEW VI, shown in Figure 6.5, to the dynamometer controller, with engine speed and throttle percentage at different time interval. Here, also the data is collected in sweeps, with speed held constant while the throttle position varied from its minimum to maximum value. The dynamometer controller is specially tuned for the transient experiment.

The engine control system use a base engine calibration in order to allow the engine to run through an individually appropriate range to prevent any damage to the engine. The base calibration was useful in the current situation as there was no constraint applied to the transient design itself. However, the required base calibration can be relatively crude and far from optimal; although the base calibration itself is a product of long conventional engineering effort. The measurement software, ATI VISION is connected to the engine control system via CAN, and record the measurement data which comprised of cylinder air charge, throttle position, manifold depression, torque and speed. This results in the total number of data for 23 levels that corresponds to an average of 460 observations.

To validate the predictive performance of the developed model, supporting validation sweeps are collected at four different level of speeds. The data collection for these level are performed at the same time, as of the main data collection to prevent any variation in the test data due to test conditions. The validation

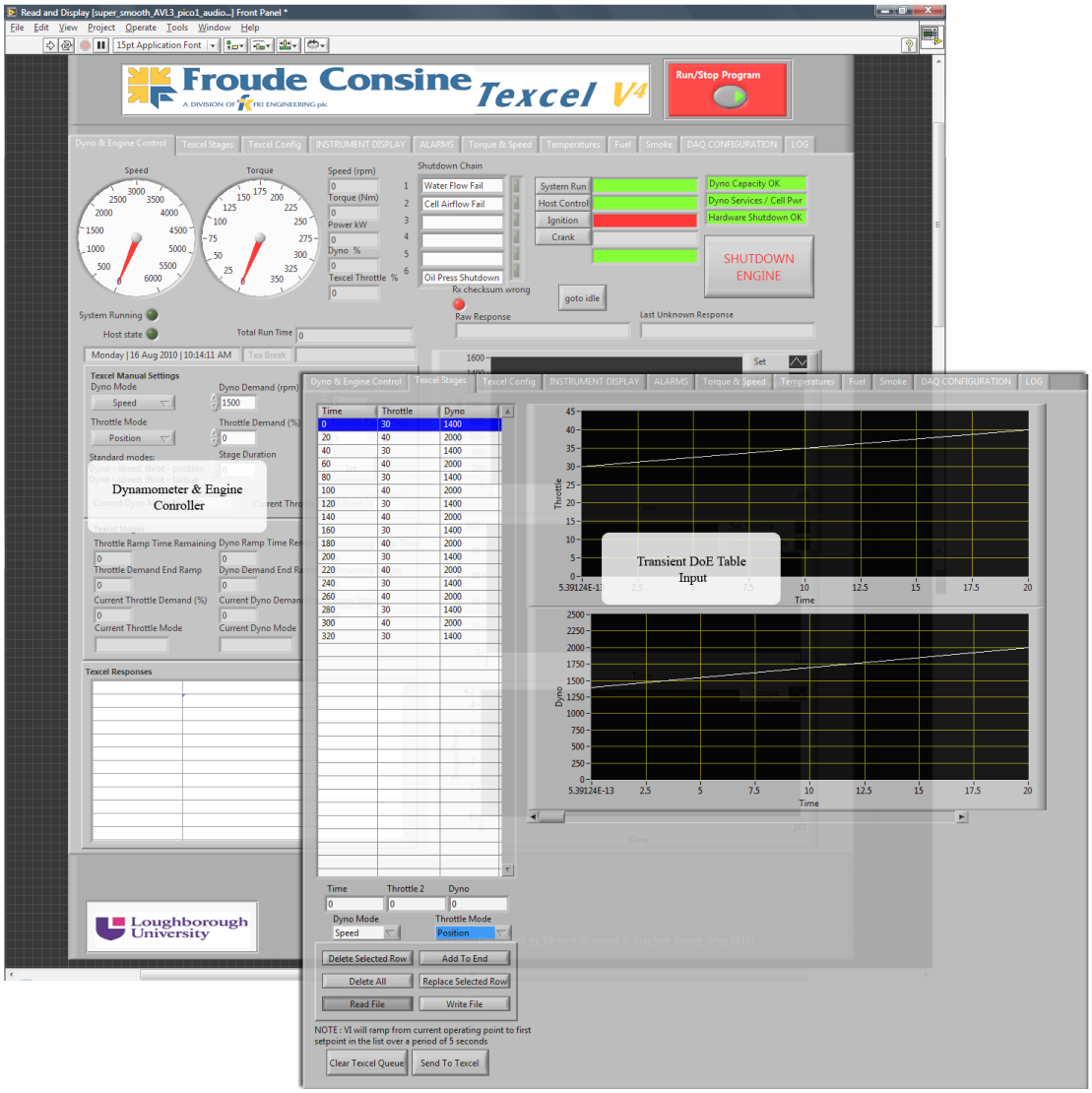


Figure 6.5: NI LabVIEW VI for Transient Engine Control

points are however, not used in the development of the transient model, but only for the measurement of the predictive capability of the developed model.

For a steady state data acquisition based of standard DoE methods only require measurement at static engine behaviour, and require no sophisticated techniques. However, the method is only limited to static problems. Whereas, for the transient purpose it is important to consider the dynamic behaviour at different

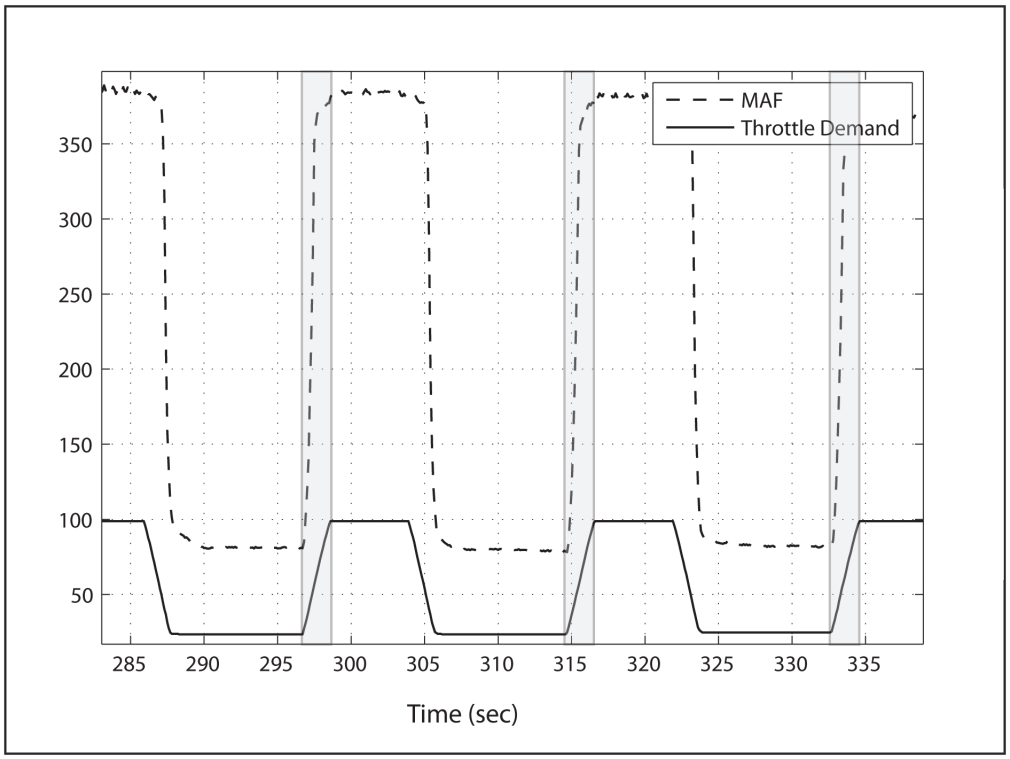


Figure 6.6: Throttle ramp for transient measurement of engine characteristics

stationary states. Therefore, the experiment has been designed to properly consider the dynamic behaviour of the engine and the controller for speed/load and temperature stability. This measurement technique stochastically cover a user-defined range of amplitude and frequencies. Figure 6.6 shows a throttle ramp for transient measurement. The throttle position is varied over time with speed held constant at each designed level during the transient measurement. However, for development of the model only throttle ramp-up data (highlighted), in Figure 6.6 and 6.7 is selected , and all other data is discarded for their irrelevance.

Postprocessing of Measured Data The raw data has to be thoroughly investigated before it could be used in the model building. This is especially important for the transient measurement data where the moving average filtering is lacking, which is commonly applied in steady state measurements. The observation at the

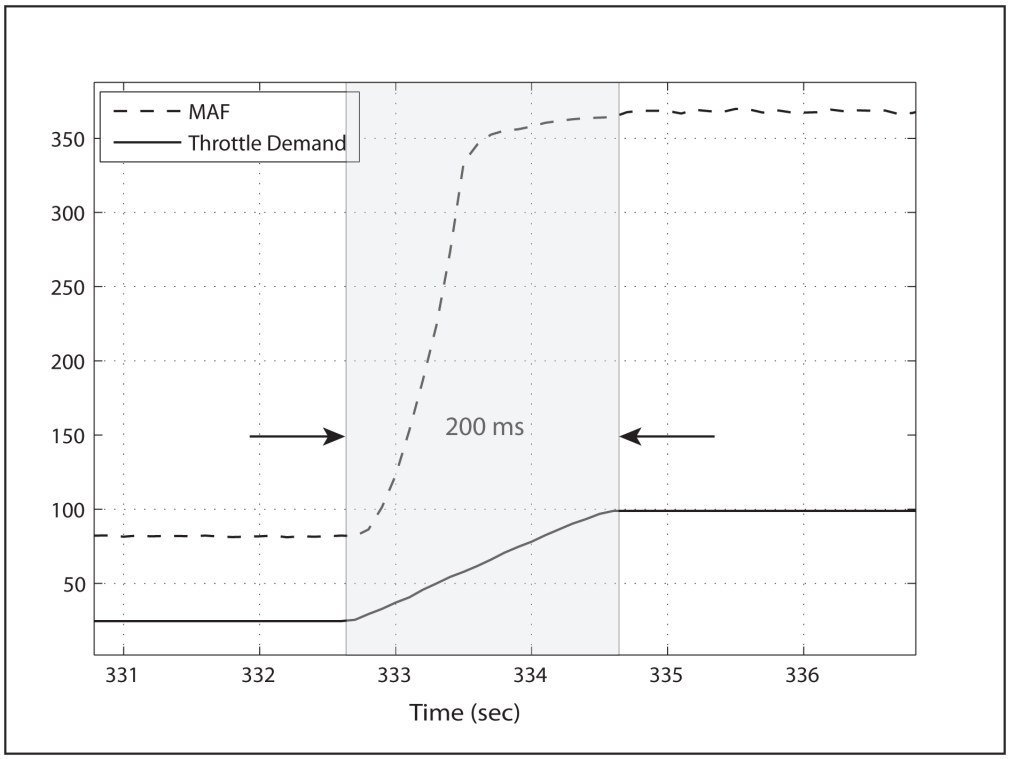


Figure 6.7: Selected throttle ramp for transient local model generation

engine speed of 900 RPM was disregarded, for the data was not suitable for the development of the model. Also, the data was checked for any outliers that would deeply worsen the data-based modelling results. The correction was preformed visually using MBC toolbox, and few data points were eliminated from different observations. These points were mostly removed from lower engine speed.

6.3 Model Structure

The model structure in block diagram form is presented in Figure 6.8. The justification of the formulation is presented in later sections, from Section 6.3.1 to 6.3.3.

In stage-1 model the relationship between the throttle and mass air flow on a

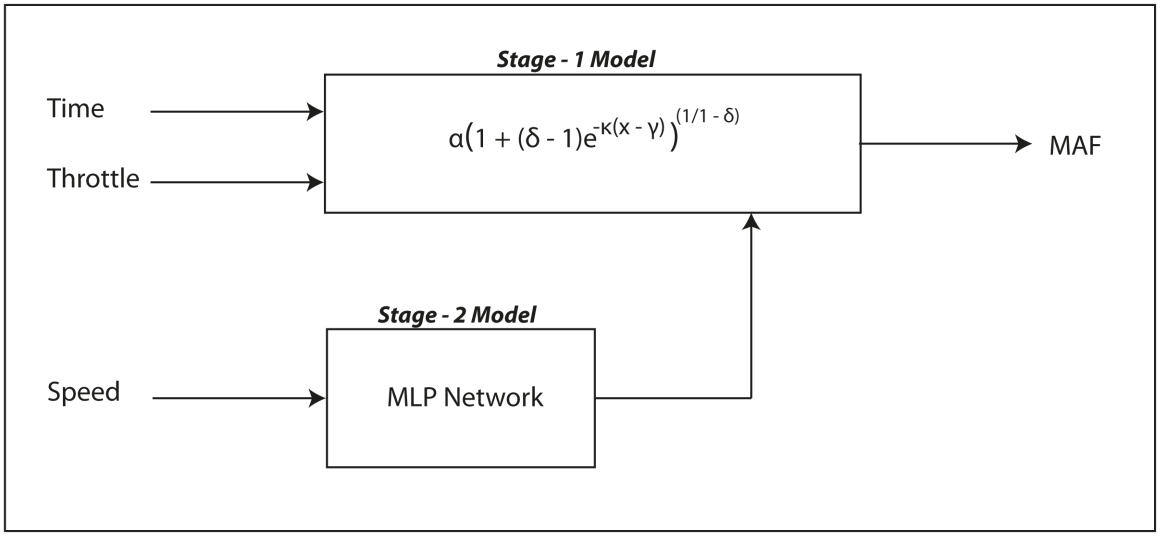


Figure 6.8: Throttle air flow dynamics model

sweep-specific basis is modelled. Richard growth model, discussed in Section 6.3.1.1 is selected and provide an adequate fit. The specification of the first model is:

$$f(x) = \alpha(1 + (\delta - 1)e^{-\kappa(x-\gamma)})^{\frac{1}{1-\delta}} + e_{ij}, \quad Var(e_{ij}) = \sigma^2 g^2(f(x_{ij}, \beta_{ij}), \zeta) \quad (6.5)$$

where $\beta_{ij} = [\alpha_{ij} \quad \gamma_{ij} \quad \kappa_{ij} \quad \delta_{ij}]^T$, with the value of regression parameter for individual i at conditions j.

The response features with good engineering interpretation regarding the location and value of the maximum value for mass air flow, as well as the location of the point of inflection on the x-axis are chosen. At second-2, a MLP network models are used to describe the systematic variation in the response features across the sweeps.

6.3.1 Local Model

At stage-1, the primary focus of the model is to find a fit function that could accurately describe the silent feature of the throttle sweep in a time domain. Local models are fitted to each test, in different places across the global space. This model is utilized to calculated the response features for the second-stage modelling.

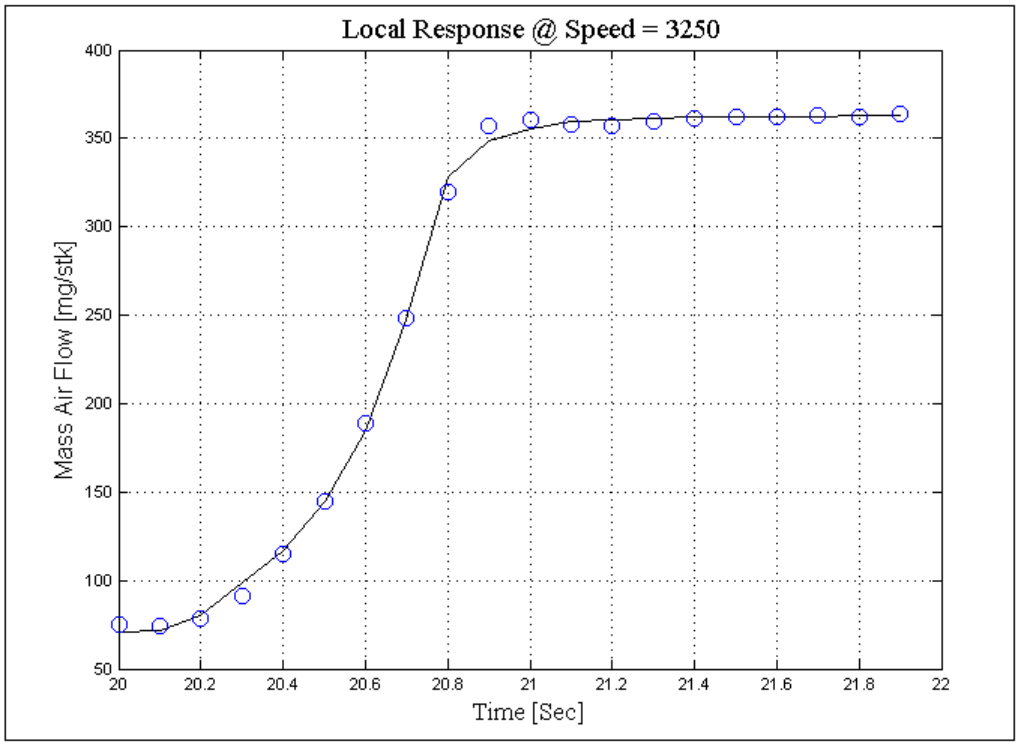


Figure 6.9: Local model for throttle air flow

The first response to model is mass air flow (MAF) against time. Analysis of models of this form is already supported by the Matlab MBC toolbox. Transient models are supported for multiple input factors, where time is one of the factors. A dynamic model is defined using Simulink and Matlab code file, that describes parameters to be fitted in this model. The detail description of the model will be describe in later sections.

Figure 6.9 shows a local response model of mass air flow verses time at an engine speed of 3250 RPM. The curve fitted increases to a maximum before it flatten off. This is because of the fact that the maximum intake pressure is achieved at the throttle angles far below the maximum value.

6.3.1.1 Richard Growth Model

The typical air flows sweep profiles, under transient operating conditions, is shown in the Figure 6.10. In this types of data, the curve does not steadily decline, but rather increases to a maximum before steadily declining to zero, behaving normally in biological growth curve by an S-shaped, or sigmoidal pattern.

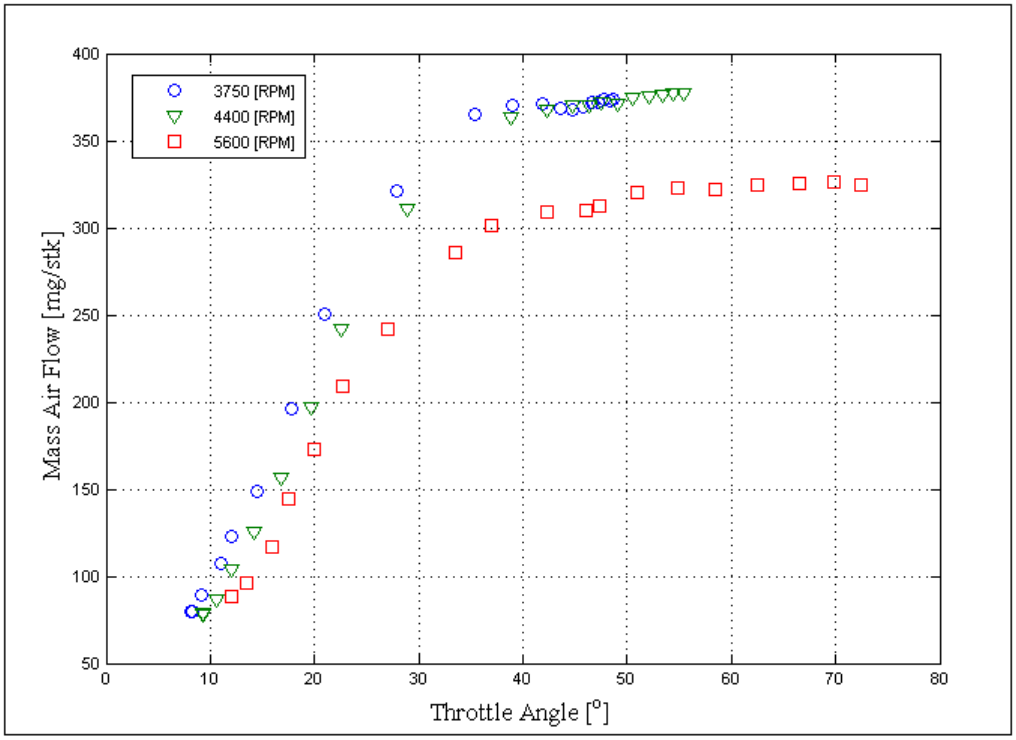


Figure 6.10: Throttle body airflow sweep profiles under transient condition

Cary (58) suggest the growth profile proposed by Richards (123; 124; 125) for

this type of data. These models had a good interpretation and have a meaningful parameters from airflow modelling perspective, and hence had an advantage over the empirical models such as polynomial equations for modelling nonlinear growth. The Richard's curve is written in mathematical form as:

$$f(x) = \alpha(1 + (\delta - 1)e^{-\kappa(x-\gamma)})^{\frac{1}{1-\delta}}, \qquad \delta \neq 1 \tag{6.6a}$$

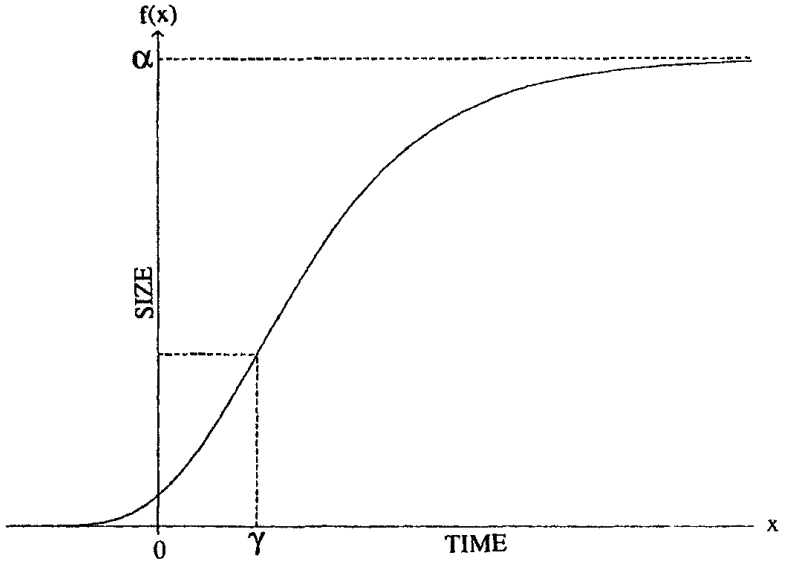
$$f(x) = \alpha(e^{-e^{-\kappa(x-\gamma)}}), \qquad \delta = 1 \tag{6.6b}$$

where α is an upper asymptote value to which the curve tends and the parameter γ locates the point of inflection on the x-axis. κ is a growth rate parameter and δ being the parameter that indirectly locate the point of inflection of the curve on the f-axis at $f = \alpha/\delta^{\frac{1}{\delta-1}}$ and thus controls the shape of the curve. The maximum growth rate is

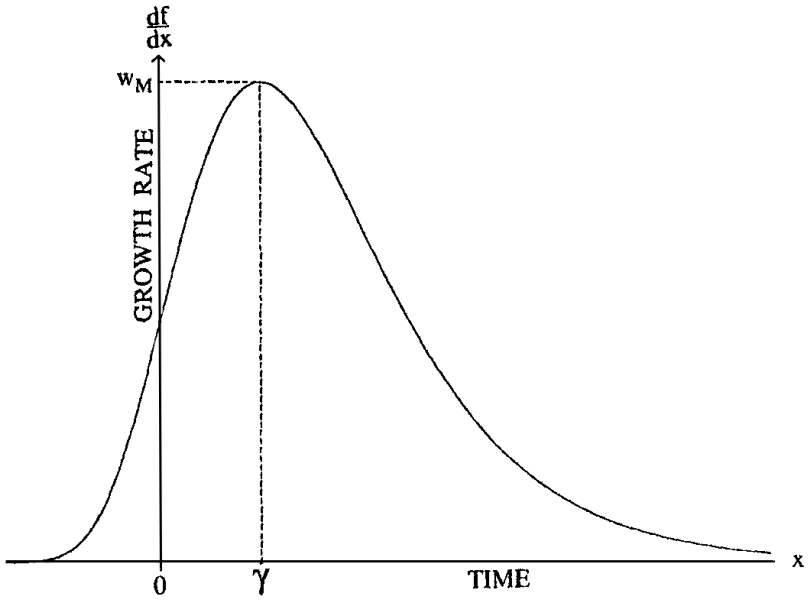
$$w_m = \kappa\alpha\delta^{\frac{\delta}{\delta-1}} \tag{6.7}$$

The Richards growth model treated δ as a free parameter provided a flexible family of curves with an arbitrarily placed point of inflection. The model includes the mono-molecular model ($\delta = 0$), the von Bertalanffy model ($\delta = \frac{2}{3}$), the logistic model ($\delta = 2$), and (by talking the limit as ($\delta \to 1$) the Gompertz equation, where growth is not symmetrical about the point of inflection.

The Gompertz, logistic, Chapman-Richards, Richard's, and the Von Bertalanffy growth models have points of inflection and are sigmoid (126), and are suitable for quantifying a growth phenomenon that exhibits a sigmoid patter over time. Lei (125) discussed that the Bertalanffy-Richards function had sigmoid and concave depends on the allometric parameter δ, and is suitable not only for representing a growth phenomenon that exhibits a sigmoid pattern over time, but also for representing a growth phenomenon that exhibit a concave pattern over time. This is the reason why this curve is selected for the current study, as it can be seen



(a) Growth curve with α = final size and γ = point of inflection



(b) Growth rate curve with w_m = maximum growth rate

Figure 6.11: Richards family of sigmoid growth model

in Figure 6.10, that the transient air flow shows a concave phenomenon at high speed.

6.3.1.2 Transient Model in Simulink/Matlab

A transient model in a Simulink and Matlab environments defined according to the equation 6.6a and 6.6b, before it can be used in level 1 of MBC environment. Figure 6.12 show the model in block form.

The model calculate the value of regression parameter $\beta = [\alpha \quad \gamma \quad \kappa \quad \delta]$. The inputs to the model are time and throttle position data, while the output of the model is mass air flow. There are two subsystems in the main system, and the free parameter δ defined the type of the subsystem to be chosen. The subsystem (a) is selected when the type of the model is monomolecular i.e., $\delta \neq 1$, while for other type of models i.e., $\delta = 1$ subsystem (b) is used.

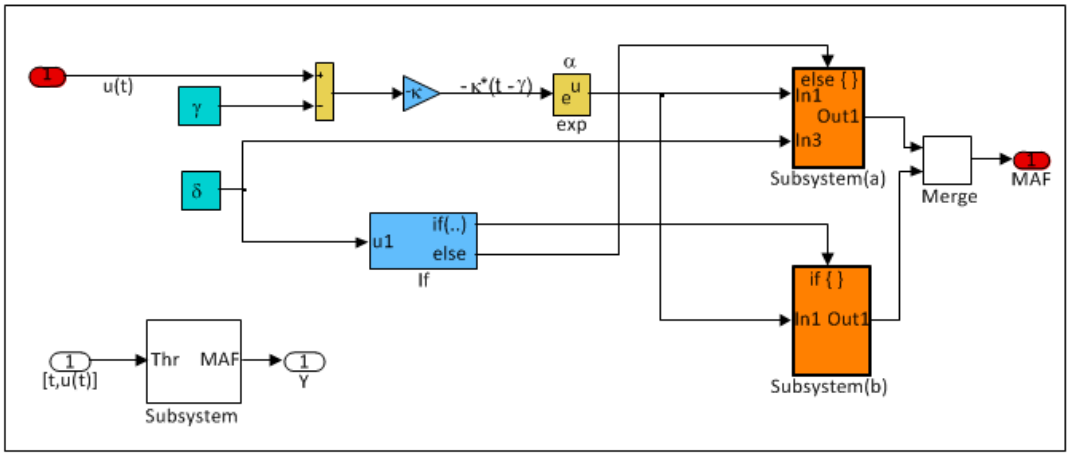
Starting Values for Fitting Richard Models The crude initial estimates of the parameters can often be obtained from a scatter plot of the growth data, perhaps with a freehand smooth curve added. For example, the final size α can often be obtained this way. The coordinates of the point of inflection, $x = \gamma$ and $f = \alpha/\delta^{\frac{1}{(1-\delta)}}$, can then provide starting values for γ and δ. Finally, the intercept on the $f-$ axis is

$$f(x) = \alpha(1 + (\delta - 1)e^{-\kappa(x-\gamma)})^{\frac{1}{1-\delta}} \tag{6.8}$$

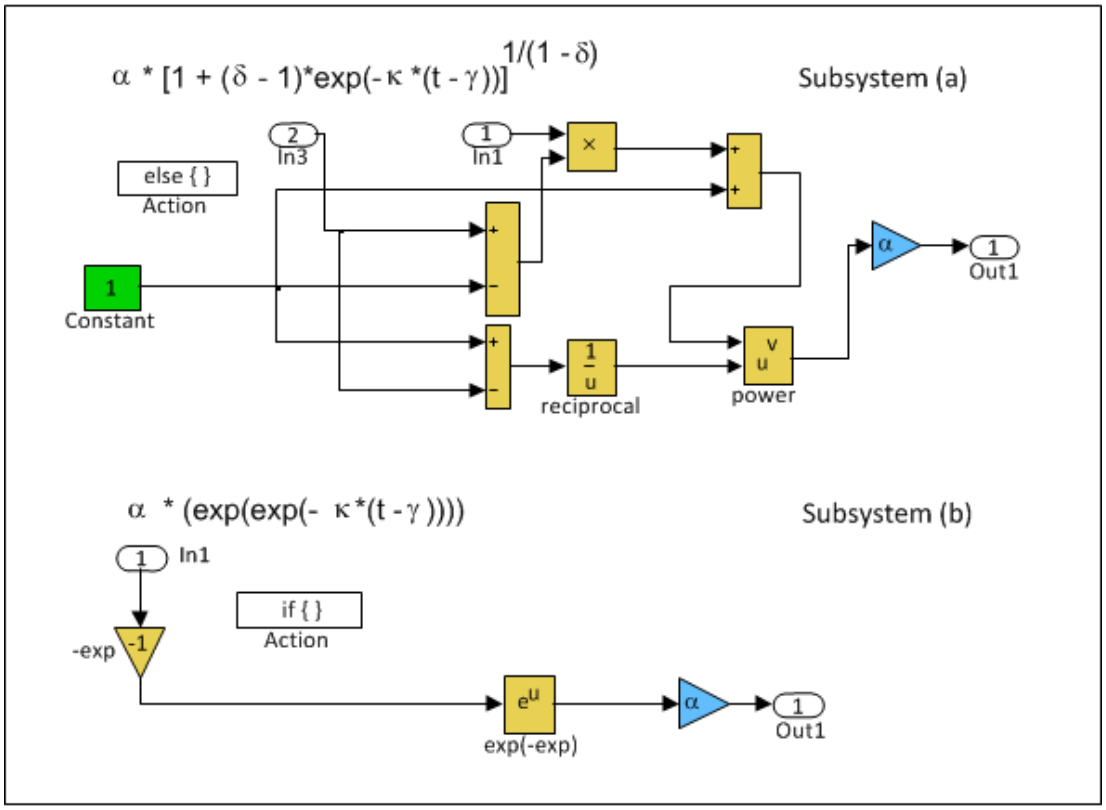
which can be solved for κ.

However, more sophisticated refinements are available if convergence cannot be obtained from the above starting values. By rearranging equation 6.6a, a transformation $f^* = g(f)$ can be obtained as

$$f^* = -log\frac{(f/\alpha)^{(1-\delta)} - 1}{\delta - 1} = \kappa x - \kappa\gamma \qquad (\delta \neq 1) \tag{6.9}$$



(a) Main system



(b) Subsystems

Figure 6.12: Simulink model for air flow dynamics

or from 6.6b

$$f^* = -log[log(\alpha/f)] = \kappa x - \kappa\gamma \qquad (\delta = 1) \tag{6.10}$$

Thus for given α and δ, starting values for κ and γ can be obtained from a simple linear regression of $y^* = g(y)$ on x, where y is the observation taken on f.

When the asymptote is not clear on the scatter plot, a range of several $\alpha-$values or a grid of α and δ values can be used with κ and γ obtained by linear regression as above for each pair (α, δ). This gives a set of possible starting values $(\alpha, \delta, \kappa, \gamma)$. The best of these can be chosen using the optimality criterion to be employed in fitting the model. Using least squares, for example, the 4-tuple $(\alpha, \delta, \kappa, \gamma)$ is chosen with the smallest value of

$$\sum_i [y_i - \alpha\{1 + (\delta - 1)e^{-\kappa(x_i - \gamma)}\}^{\frac{1}{1-\delta}}]^2 \tag{6.11}$$

Richards described a simple method for estimating a for the monomolecular model ($\delta = 0$) using three equally spaced time points $x_1 x_2, and x_3$, which should be widely separated. Let $\tilde{f}_i$ be the estimate of $f(x_i)$ obtained visually from a freehand curve through the scatter plot of y versus x. Then from equation 6.6b it is easily verified that

$$\frac{\alpha - \tilde{f}_2}{\alpha - \tilde{f}_1} = \frac{\alpha - \tilde{f}_3}{\alpha - \tilde{f}_2}$$

so that

$$\alpha = \frac{\tilde{f}_3^2 - \tilde{f}_1\tilde{f}_3}{2\tilde{f}_2 - \tilde{f}_1 - \tilde{f}_3} \tag{6.12}$$

For any $\delta \neq 1$ the same formula applied to the $\tilde{f}_i^{1-\delta}$ gives an estimate of $\alpha^{1-\delta}$. For the Gompertz curve, the method applied to $log\tilde{f}_i$; gives an estimate of $log\alpha$. Bad

ill-conditioning and convergence problems have often been experienced in fitting the Richards model to data. Both problems can occur when insufficient of the curve is visible to provide a good initial estimate of the final size α. Data which give rise to only moderate ill-conditioning when logistic or Gompertz curves are fitted often give rise to pronounced ill-conditioning when the full four-parameter Richards model is fitted. Curves with quite different $\alpha-$values look very similar, and difficulty in distinguishing between them with even a small amount of scatter is anticipate. Indeterminacy in δ or the location of the point of inflection on the $f-$axis (at $x = y$) leads to indeterminacy in y. The parameters δ and κ are also tied together through the fairly stable growth-rate parameter $\lambda = \kappa/[2(\delta + 1)]$.

6.3.2 Response Feature Selection

The local model fitted to the data in Figure 6.9 shows two distinct point; the upper asymptote α and point of inflection on x-axis γ in the curve which can be interpreted directly as a response features. The response feature α would increase with the increasing engine speed, that is easily explained by considering the nature of a typical naturally aspirated wide-open throttle (WOT) brake torque characteristic for an engine with fixed cam timing (4). In the mid speed range, it is at the maximum of its value. The rate of airflow decrease with the increasing speed at high speed, because of the the drop of intake manifold pressure at WOT than atmospheric due to the increase in flow losses. However, these arguments would differ by the existence of intake tuning phenomenon at specific speeds.

Similarly, the profile for γ would increase rapidly with engine speed, and the flatten off at the higher speed. This is because of the fact that the at low engine speed maximum intake pressure is achieved at the throttle angles far below the maximum value. The point of inflection for the curve should correspond to a lower throttle angle at reduced speed.

The parameter δ indirectly locates the point of inflection of the curve and control the shape of the response function. Therefore, can be treated as a response feature. However, Richard (124) derived an equation for an average growth rate;

$$\frac{1}{\alpha}\int_0^{\alpha}\frac{\kappa f}{1-\delta}\left[\left(\frac{f}{a}\right)^{\delta-1}\right]df = \frac{\alpha\kappa}{2(\delta+1)} \tag{6.13}$$

Thus for the final size, $\kappa/[2(\delta+1)]$ $(=\lambda$, say) is a crude measure of *"growth rate"* for comparisons between different sweeps, with average growth increases with increasing engine speed. Also, κ acts as a scale factor on time x, so that for fixed δ it acts as a rate parameter.

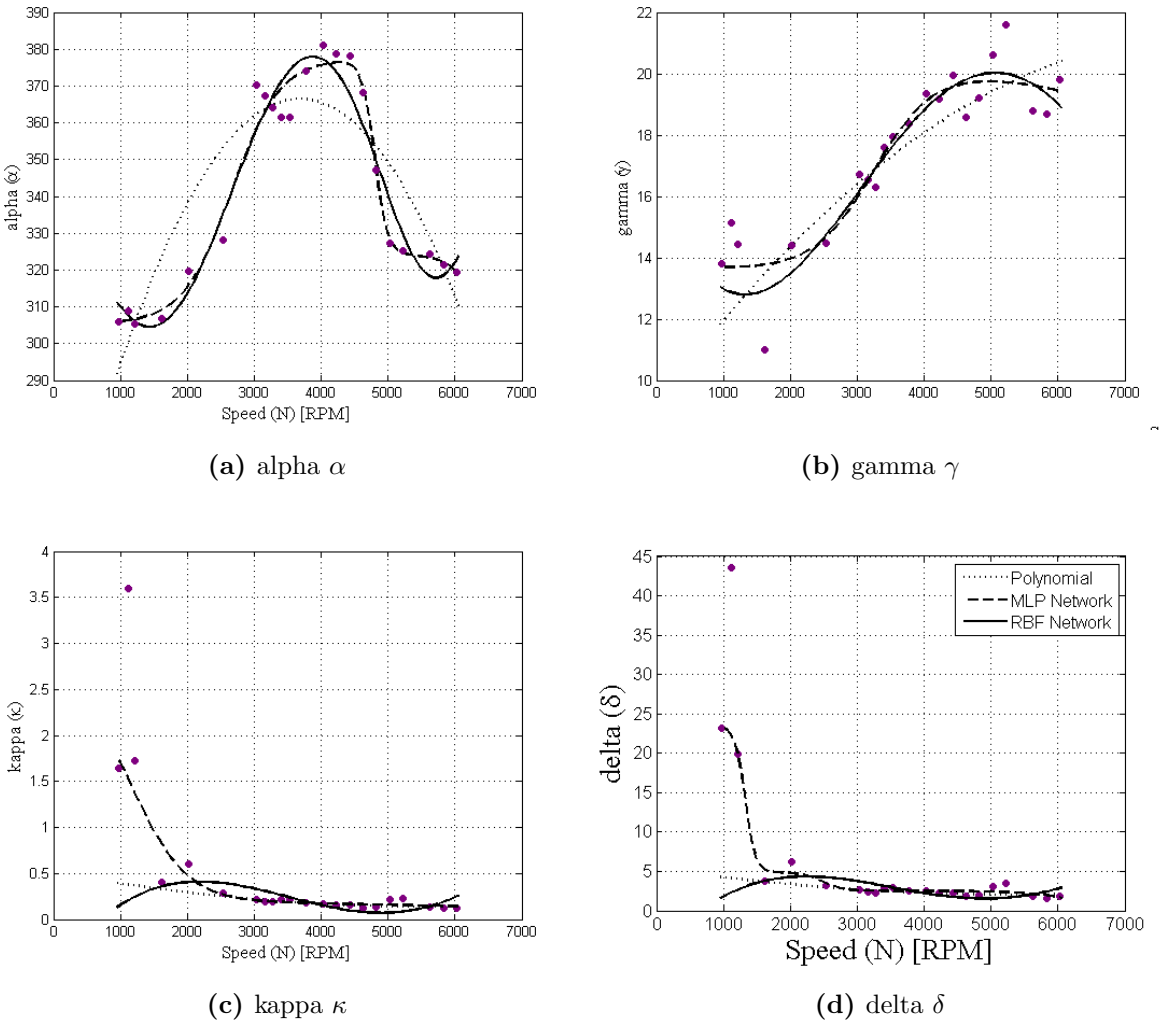
6.3.3 Global Model

The response features are modelled in the stage-2 as a function of engine speed to explain the variation across the sweeps. Three different modelling methods, i.e., Polynomial, Radial basis function (RBF) and Neural Network (NN) are used to model the systematic variation in the response features. These models are constructed as;

- A second degree polynomial model is used (i.e. $n = 2$). As the order of a polynomial increases, it is possible to fit more and more turning points. The curves produced can have upto $(n-1)$ bends for polynomials of order n. However, higher-order polynomial have an embedded tendency toward over-fitting, that should be avoided while fitting any curve.

- The multi-layer perceptron network is fitted with Bayesian regularization method, *trainbr*, that updates the weight and bias values according to Levenberg-Marquardt optimization. The architecture of the MLP network contain one hidden layer, with 3 number of neurons in the layer for each response feature.

- A radial basis function model is fitted using Regularised Orthogonal Least Squares (ROLS) (127) technique. The model algorithm is initialised with approximately 25% of the total available data points, in order to achieve a good fit and at the same time avoid over-fitting.



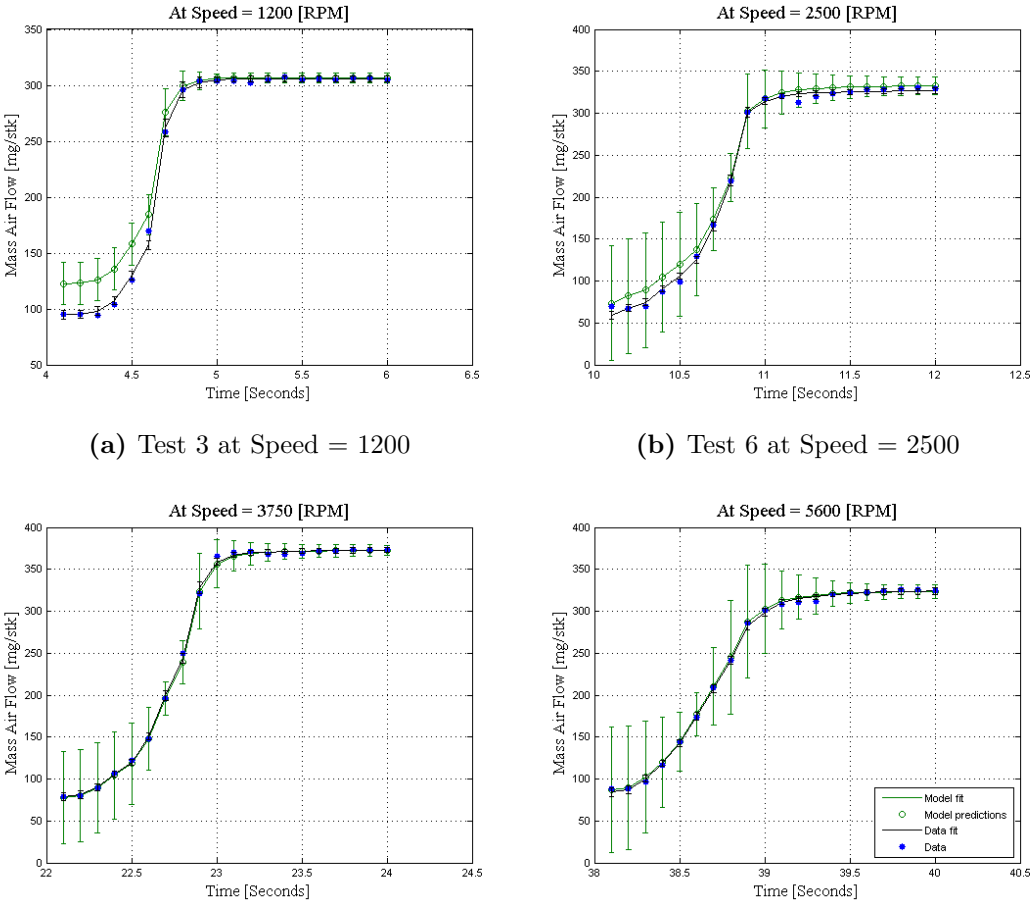
(a) alpha α **(b)** gamma γ

(c) kappa κ **(d)** delta δ

Figure 6.13: Comparison of response feature trend analysis for three different model used

The corresponding analysis is shown in the Figure 6.13, which reveal the response features trend with respect to the second stage covariate, the engine speed. The response feature α is increase with increase engine speed, which a typical for a naturally aspirated wide-open throttle brake torque characteristic for an engine with fixed cam timing. It has a maximum value at mid speed range, due to high air induction. The point of inflection γ correspond to a lower throttle angle at

reduced speed. It increase rapidly with engine speed, with a flatten trend towards the end.



(a) Test 3 at Speed = 1200

(b) Test 6 at Speed = 2500

(c) Test 12 at Speed = 3750

(d) Test 20 at Speed = 5600

Figure 6.14: Training data fit to various test sweeps

A visual inspection of the response feature curve trend in Figure 6.13 shows a very complex structure. The simplest polynomials models is having very little use, where it unable to fit most of the features in the global model. The neural network models does a great job in estimating the fit parameters. These parameters often have a physical interpretations, a major aim of the investigation is to estimate the parameters as precisely as possible, and a further aim is to test

the fit of the data to the model. The modelling itself is a trade off - too few parameters means the shape of the surface cannot be captured, while too many parameters gives a risk of overfitting.

Figure 6.13 shows comparison of the three different models used. It is clear that the polynomial model failed to show the different features in the data and do not fit the model to the data well. The RBF model, however shows a really good improvement than the polynomial model, with increase number of the parameter fits and with the capture of response feature really well.

However, the MLP network model tend to show a strong relationship to the fit of data and also avoiding the data points for unnecessary risk of overfitting. The number of parameters in case of MLP models is increased to double as compare to the RBF model. The parameters serves to relate functions and variables using a common variable when such a relationship would be difficult to explicate with an equation. These parameters often have a physical interpretations, a major aim of the investigation is to estimate the parameters as precisely as possible, and a further aim is to test the fit of the data to the model. The increased number of parameters permit the possibility of fitting the response feature with great accuracy, with the same number of observations as for the other model choice. This reinforce the previous discussion in Section 5.2.4, for the preference of using the MLP neural network model over the RBF model.

Finally, the two-stage transient engine model is examined by comparing it with the local fit and with the data. The local air charge/time curve is reconstructed by taking the values of corresponding response feature curvatures from the two-stage model, and then confirm this reconstructed curve against the original fit and the data. Figure 6.14 plots the corresponding response surface for air charge as a function of time. The developed model regression summary is presented in Table 6.2. The response model is well represented by MLP network, with the two-stage RMSE of 10.1517 [mg/stk], compare to polynomial and RBF model having two-stage RMSE of 15.0769 and 31.3908 respectively.

Table 6.2: Regression summary statistics

Response Model	Model Type	Local RMSE	Two-Stage RMSE	Two-Stage T^2	Validation RMSE
MAF	Polynomial	4.1383	15.0769	3.5634	20.5551
	RBF	4.1383	31.3908	2.4991	15.4398
	MLP	4.1383	10.1517	2.5235	14.0596

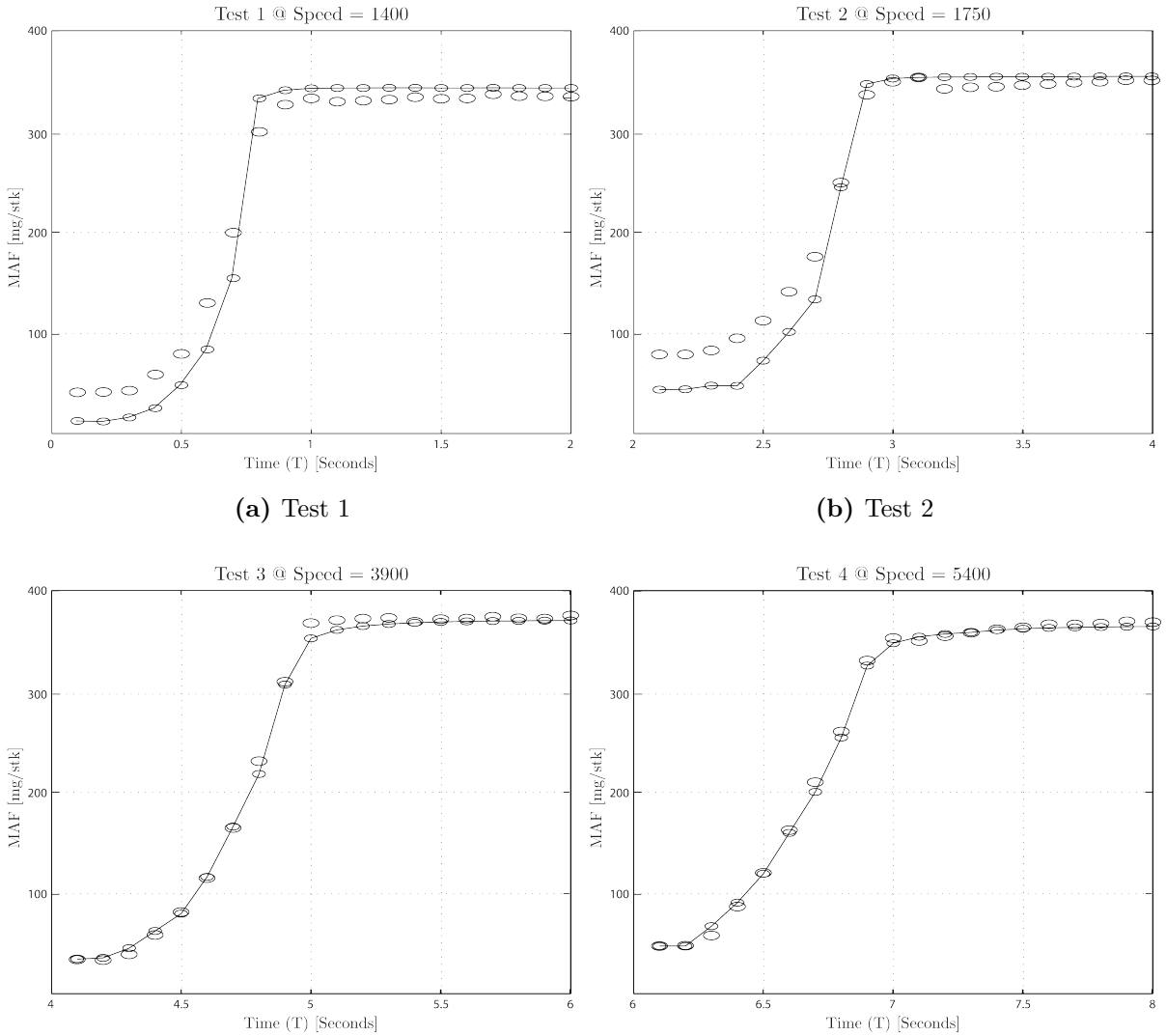
Table 6.3 summarise the comparison of three different model types used. Their is a clear difference between the neural network based models in terms of RMSE, with MLP and RBF network models having the errors of 10.1517 [mg/stk] and 31.3908 [mg/stk] respectively. The polynomial model having RMSE of 15.0769 [mg/stk]. Also, the number of parameters fit in the MLP network compare to the RBF network is double, which mean the degree of accuracy increases as the number of parameters increases. These trends are also clear in Figure 6.13 where polynomial model unable to fit most of the regression parameters. The RBF model under fit in κ and δ, and could not show the required trends. However, the MLP model has clearly good fit among these models.

6.4 Developed Model Validation

In addition to the data collection for the model development, a further four throttle sweeps were collected to serve as a small validation data pool. These data were collected at the same time as the training data, to ensure the consistency and avoid any error due to variation in the test conditions. These data were not used in training the model. The model validation plots are shown in Figure 6.15. As discussed three different model are proposed to describe the response features behaviour at various engine speeds. The neural network based model fit is shown here, where it provide good predictive capability.

Table 6.3: Models fit summary statistics

Response Feature	Model Type	Observation	Parameters	RMSE	BIC	AIC_c
Alpha (α)	Polynomial	22	3	13.497	123.649	121.638
	Radial Basis Function	22	5	7.641	102.282	101.298
	Multi-Layer Perceptron Network	22	10	6.083	100.110	114.508
Gamma (γ)	Polynomial	22	3	1.312	19.080	21.092
	Radial Basis Function	22	5	0.981	9.865	8.077
	Multi-Layer Perceptron Network	22	10	1.264	30.991	45.389
Kappa (κ)	Polynomial	22	3	0.490	-22.208	-24.219
	Radial Basis Function	22	5	0.453	-24.527	-26.393
	Multi-Layer Perceptron Network	22	10	0.173	-53.665	-35.822
Delta (δ)	Polynomial	22	3	6.068	88.442	86.431
	Radial Basis Function	22	5	5.251	83.664	81.871
	Multi-Layer Perceptron Network	22	10	0.833	12.241	30.885



(a) Test 1 **(b)** Test 2

(c) Test 3 **(d)** Test 4

Figure 6.15: External validation

The validation root mean square error of prediction using the validation data is 14.0596 [mg/stk] for MLP network compare to 15.4398 [mg/stk] for RBF network model.

6.5 Summary

This chapter has presented a novel method for transient engine modelling for calibration using two-stage regression approach. The method is illustrated with a simple example for accurately modelling the transient throttle body air flow characteristics. The objective was to support subsequent calibration of look-up table formulations for the transient air flow response surfaces characteristics. Two-stage sweep specific data methodology was used to collect data for regression parameters estimation.

At stage-1 a Richard growth curve was fitted to the data, which show suitable not only for representing a growth phenomenon that exhibits a sigmoid pattern over time, but also for representing a growth phenomenon that exhibit a concave pattern. The local model at stage-1 were shown to provide air flow characteristic with throttle angle in first stage.

At stage-2 MLP network was again used to describe the behaviour of the response features with the engine speed. Also, a polynomial and RBF models were generated for comparison and were fitted to these response features. Models were selected on minimising a RMSE criterion.

The objective of the throttle body air flow modelling was to support the calibration of the look-up table indexed by engine speed and throttle position at different interval of time. This look-up table is utilized in the control strategy for fuel metering purposes during back flow or in MAF sensor failure, at transient conditions.

The corresponding calibrations are developed in a later chapter.

7

Application of Transient Engine Model in Calibration

The previous chapter demonstrated the development of transient engine model using two-stage regression approach for the analysis of structured data collected during the course of engine mapping studies. Once the modelling stage is competed the focus turns to the generating of an appropriate engine calibration.

In this chapter, a case study is presented for the application of the model for the generation of lookup table for the throttle body air flow indexed by engine speed and throttle position at different interval of time. This look-up table will be utilized in the control strategy for fuel metering purposes during back flow or in MAF sensor failure, at transient conditions.

7.1 The Characterisation Problem

The focus of the characterisation problem is the application of developed transient engine model to engine calibration. As discussed in Chapter 1, the fundamental response surfaces (*e.g.* indicated torque or MBT spark-timing) representations are embedded within the control strategy as algebraic combination of one and two dimensional look-up tables, each indexed by the appropriate state or actuation

variables.

First stage of the model-based calibration development process involves transient engine modelling that can capture full engine dynamics across a wide range of transient time scales. The approach requires a range of timescales to be captured in the modelling, which in turn require that the underlying data contain those transient features. As an example, for any particular engine speed and operating torque, there are a number of potentially feasible combinations of say injection timing and EGR rates that give rise to the same NOx levels. Thus a choice has to be made in real-time between these competing controlling parameters.

The second part of the implementation of the model-based technology involves the development of the strategy for the controller. The controller utilizes specific desired engine performance, emissions and fuel efficiency outputs to dictate the required engine control input that in turn result in those particular outputs. This approach constitutes a situation where the fundamental response surfaces representations are embedded within the control strategy as algebraic combination of one and two dimensional look-up tables. The approach is very simple as compare to the (43; 44) which uses complex inverse model for development of new controller as an alternative to conventional map or table-based methods, which require modification in engine hardware or an entirely new engine controller.

7.2 Engine Calibration Generation

The control algorithms are fixed and therefore the response models derived have to be fitted to the appropriate control feature response. The control algorithms for a particular feature may rely upon a set of equation coefficients or a series of two or three-dimensional tables. The memory allocation reduces significantly by regressing a model to its equation coefficients, however the disadvantages are that it may be difficult to regress a response surface into a containable linear equation (36).

Lygoe's (128) discusses the fitting of control look-up tables to a response surface model. The paper illustrate the relationship of algebraic combinations of look-up tables to the corresponding components of the reference surface model, the optimal selection of the *break points*. These points are used to minimise the integral squared difference between the projection of the model on the look-up table and the surface defined by look-up table itself. The strategy generates predictions between table points using bi-linear interpolation, which results in errors between the two surfaces. If the errors are large, then quite severe performance limitations may occur.

If the look-up table representation is rarely capable of approximation of the true behaviour of the response feature over the entire region of operability, then errors between the strategy and reference model also arise. This can be reduced by utilising the model based calibration methodology, where a model can be used to derive an improved representation of the response characteristic in strategy.

The use of advanced model fits utilising splines or neural networks witnesses a significant increase in the number of terms. Moreover, once in-vehicle work commences the calibration may require subtle altering to suit in-vehicle conditions. Presenting a calibrator a large number of coefficients thereby render it near impossible for in-vehicle calibration changes due to the fact that each coefficient may affect many areas of operation. It is therefore beneficial to utilise a series of calibration tables, these have to be large enough to contain the complex surface shapes but not excessively so to require significant memory allocation. A generic tabular strategy may be represented by the following:-

$$\begin{aligned} \text{X Response} &= \text{Table A (Speed, Load) + Table B(Speed, Coolant Temp)} \\ &= \text{+ Table C (Lambda) + SCV Modifier + } \ldots \end{aligned}$$

7.3 Calibration in CAGE

Once the transient engine model is setup, and validated against the external data, it can be used to setup a calibration using 'CAGE' (Calibration Generation). CAGE is the Mathworks tool which interacts with the MBC toolbox to allow the interrogation of the models generated (121). Several option are available for calibration and optimisation of the engine models. However, for the purpose of application of transient model the 'feature calibration' tool is used.

A feature calibration is the process of calibrating lookup tables and their normalizers by comparing an ECU strategy (represented by a Simulink diagram) to a model. The collection of algebraic lookup tables is determined by a strategy. It is used to estimate signals in the engine that cannot be measured and that are important for engine control. An electronic control unit (ECU) subsystem is calibrated by directly comparing it with a plant model of the same feature.

A feature calibration has an advantage compared with simply calibrating using experimental data. Data is noisy (that is, there is measurement error) and this can be smoothed by modelling; also models can make predictions for areas where there is no data. This means the engine can be calibrate more accurately while reducing the time and effort required for gathering experimental data.

7.4 A Case Study Description:

The purpose of the transient airflow modelling in previous chapter was to support calibration of the main inferred airflow look-up table **FNTHROTTLE** with an additional degree of freedom as time. This failure mode look-up table (FNTHROTTLE) indexed by throttle angle (θ) and engine speed (N) in the control strategy determined the amount of air flow through the throttle body in a conditions where backflow is indicated or in areas of high pulsation. However, the table contains the throttle body air flow at a barometric pressure of 29.92 [inHg] and ambient temperature of 100 [°F] at a steady stage condition (58). The table ignore the transient mass airflow phenomenon and therefore affects fuel metering.

A feature refers to the object that contains the model and the collection of lookup tables. A simple feature for calibrating the lookup tables for the transient airflow consists of:

- A model of transient mass airflow.
- A strategy that adds the two following tables:
  - A speed (N), Throttle (Thr) table
  - A table to account for the behaviour in Time (T)

7.4.1 Transient Airflow Model Description

A feature model is created in the first step, with assigning a response model created in chapter 6 to it. Figure 7.1 shows a three-dimensional response characteristic of throttle body air flow that has been developed and validated previously. At lower engine speed, the airflow is a sigmoid-like profile versus throttle. There is a sharp increase in an airflow at throttle angle even well below the maximum, while a relatively flat surface after that.

In the mid speed range, it is at the maximum of its value. The rate of airflow decrease with the increasing speed at very high speed, because of the the drop of intake manifold pressure at wide open throttle (WOT) than atmospheric due to the increase in flow losses. Also, the profile of high engine speed verse throttle is more concave than the sigmoid, as it is in low engine speed.

This response model across the whole time range is used to produce a final calibration table.

7.4.2 Model Strategy

Considering the response features for airflow in Figure 7.1, a strategy is created. A strategy is an algebraic collection of tables, and forms the structure of the

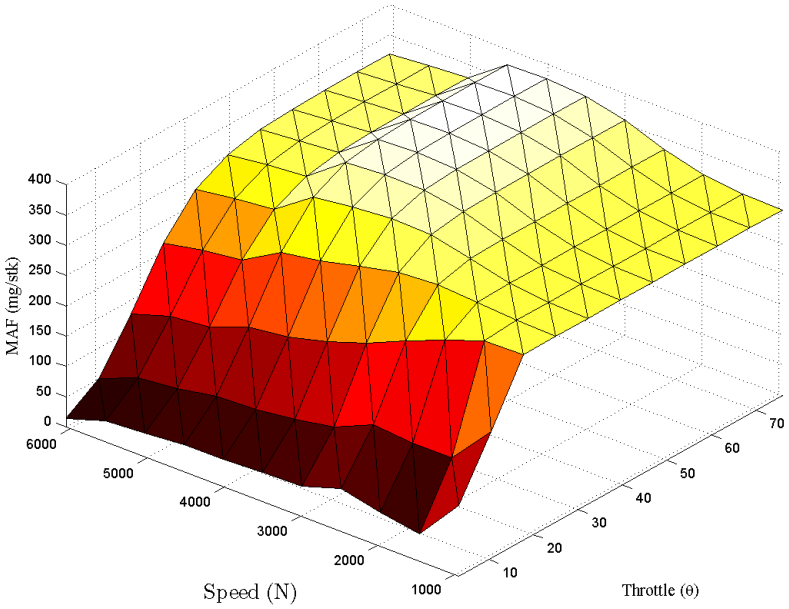


Figure 7.1: Throttle body airflow response surface

feature. The strategy for the calibration generation resembled the following:

$$\text{MAF} = \text{Fn (Speed, Throttle Position, Time)} \tag{7.1}$$

Hence, a simple strategy to calibrate a feature for transient MAF adds two tables:

- A table ranging over the variables speed and throttle
- A table to account for the behaviour of the model as the time varies.

To evaluate the feature side by side with the model, a strategy should takes some or all of the same variables as the model. The strategy is expressed using a Simulink diagram in Figure 7.2, showing a 2D lookup table for speed and throttle variables, with a 1D table for the time variable. These two tables are added together to calibrate a feature for mass air flow.

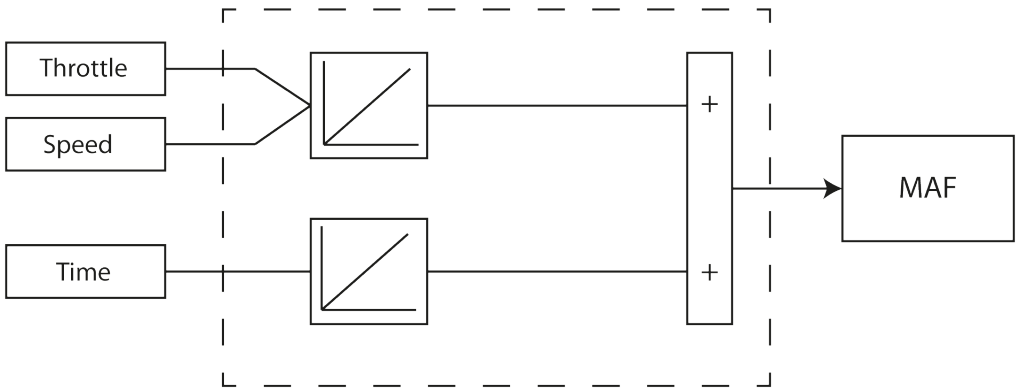


Figure 7.2: Airflow strategy for ECU subsystem

7.4.2.1 Calibrating the Normalisers

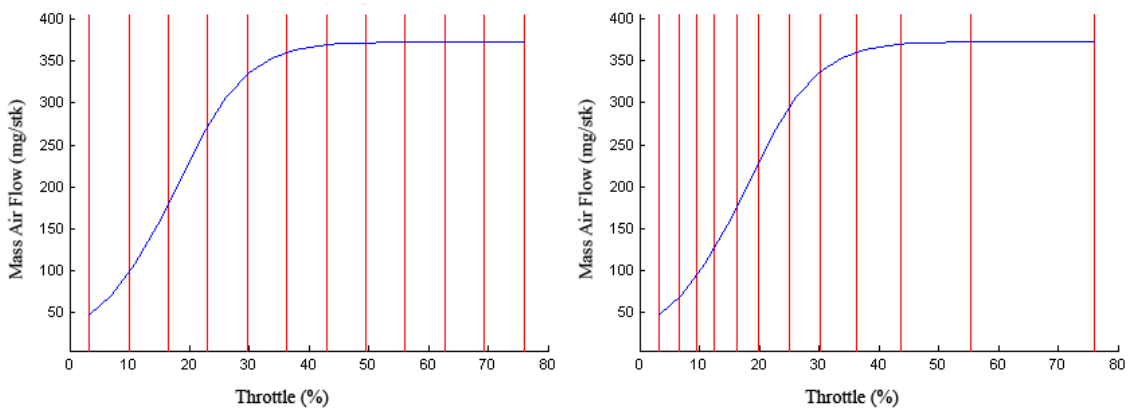
The main aim of calibration generation is to achieve the best fit of a look-up table to a model. And is achieved by minimising the maximum interpolation error between the response surface model and the lookup table adjusting the *breakpoint* position. These breakpoints are the number of look-up table rows (columns) (128). A table for the speed and throttle is set to have 10 rows and 10 columns.

Initialising and Filling Breakpoints The breakpoints at even intervals along the range of the variable are initialise for the normaliser. These normalisers are the axes for the lookup tables. The normalisers automatically initialises the table by spacing the breakpoints evenly over the ranges of the selected input variables.

For a MAF table with two normalizers of engine speed and throttle, the breakpoints are spaced for both normalisers over the range 950 rpm to 6000 rpm for speed and 4 to 75 percentage for the throttle sweep.

Filling breakpoints spaces the breakpoints by reference to the model. In feature calibrations the majority of the breakpoints are placed where the curvature of the model is greatest (shown in Figure 7.3).

For example, a model of the spark angle that produces the maximum brake torque (MBT) has the following inputs: engine speed N, relative air charge L, and air/fuel ratio A. The breakpoints are spaced for engine speed and relative air charge over the range of these variables by referring to the model.



(a) Breakpoints initilised **(b)** Breakpoints filled

Figure 7.3: Initilisation and filling of breakpoints

CAGE spaces the breakpoints by reference to the model, at selected points of the other model variables. For the case study, the input variables are engine speed N, throttle Thr over a range of time T. The model have 2 number of points over the range of time. These number of points means that it takes two slices through the model at minimum and maximum value of T. Each slice is a surface in N and Thr. That is, MAF(N, Thr, minT) and MAF(N, Thr, maxT). The average value of these two surfaces is computed to give an average model MAFAV(N, Thr).

Optimising Breakpoints The main aim in the feature calibrations is to optimise breakpoints to alter the position of the table normalisers so that the total square error between the model and the table is reduced. This routine improves the fit between the strategy and response model. The breakpoints are moved to reduce the peak error between breakpoints.

The strategy employs bi-linear interpolation to determine the desired output between table cell sites. Figure 7.4 is a one-dimensional lookup table illustrate the importance of breakpoints locations, and shows how the optimisation of breakpoints positions can reduce the difference between the model and the table.

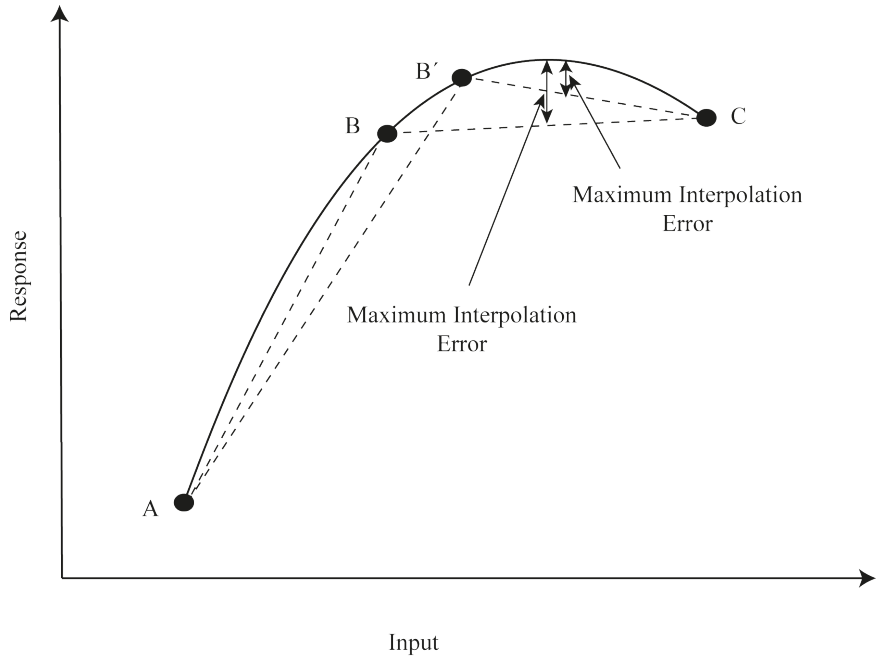


Figure 7.4: Effect of breakpoint selection on maximum interpolation error

The desired characteristics are depicted in black. Consider the breakpoint sequence A, B, C. The approximate location and magnitude of the maximum interpolation error for the interval (B, C) is also portrayed. If the breakpoint B is moved to a new location of B', the associated interpolation error in the same interval is decreased substantially. There is an obvious implications of breakpoint position for the 2-dimensional case.

Lynoe (128) has suggested minimising the integral square error over the table domain as a mechanism for selecting optimal breakpoint positions. In his study a polynomial response surface representation was employed, and integration of the error surface over the space spanned by the table axes was easily accomplished.

Cary (58), however replaced integration by summation over a uniform evaluation grid, which has much higher resolution then the lookup table. He suggested 6 to 10 times as many subdivisions per axis, or equivalently 36 to 100 times as many cells, provides a good trade-off between computational speed and precision.

The effect of optimising the breakpoint is shown in Table 7.1. The reduction in the error statistics is very obvious, and there is 90% drop in the error by optimising the breakpoints.

Error Statistics for Graph	Non-optimised	Optimised
Maximum Absolute Error	23.14	3.62
Maximum Square Error	8.4	0.82
Total Square Error	3360	326.3

Table 7.1: Difference between optimising breakpoint

The grid is defined in the optimisation process, and combined using cubic splines to approximate the model. Then the table filled with the mesh is calculated at the breakpoints. Then CAGE calculates the total square error between the table values and the mesh model.

7.4.2.2 Calibrating the Tables

A table is defined to be either a one-dimensional or a two-dimensional lookup table. One-dimensional tables are sometimes known as characteristic lines or functions. Two-dimensional tables are also known as characteristic maps or tables.

Each lookup table has either one or two axes associated with it. These axes are normalizers.

Initialising and Filling Table Values Initializing table values sets the value of every cell in the selected table to a constant. However, these values should be initialise by keeping in view about the strategy. In the study, the Time table is used a modifier and added to a single speed-throttle table to adjust for the effect of different time interval on the MAF output.

If the table is a modifier that is added to other tables, it is initially filled with zeros; if it is a modifier that multiplies other table, it is filled with 1s.

The table values are filled and optimise by reference to the model, and multiple tables at filled in this process. CAGE offers two type of filling methods for lookup tables; filling table by extrapolation and filling table by interpolation.

Optimising Table Values The table values are optimise to minimise the current total square error between the feature values and the model. The fit between the strategy and the model is optimise, by shifting the cell values up and down to minimise the overall error between the interpolation between the model and the strategy. In this process, the constraint that the model and strategy must agree exactly at the coordinates defined by the indexing scheme is relaxed, which results in the reduction of interpolation errors.

The difference in optimising table values is compare with illustration in Figure 7.5. When the table value at A and B are perturbed to A' and B', the maximum interpolation error is reduced. In this case, the table value will not be in exact agreement with the reference response surface model at the coordinates specified by the corresponding index scheme. Optimal table filling is accomplished by minimising an appropriate least squares cost function evaluated over a high-resolution user-specified evaluation grid.

The evaluation grid is equally spaced over the space spanned by the table axes. Again, here Cary (58), suggested 6 to 10 times as many subdivisions per axis, or equivalently 36 to 100 times as many cells, provides a good trade-off between computational speed and precision.

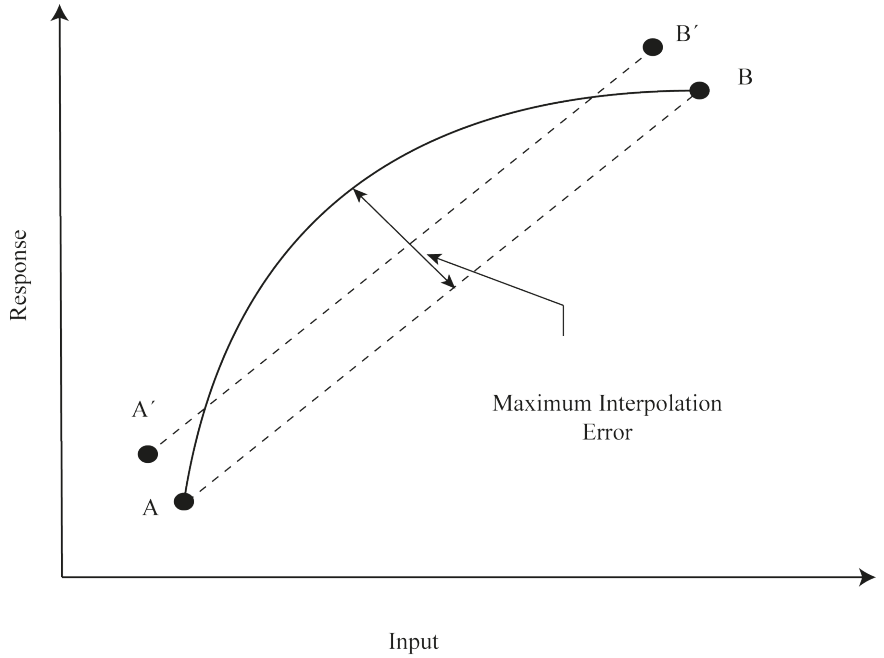


Figure 7.5: Effect of optimising table lookup values on maximum interpolation error

The effect of evaluation grid size is illustrated in lookup Table 7.2 and 7.3. Both of these table are setup at time 10 sec, with the first table created at the grid size equal to the number of cells in the table, while latter at 6 times subdivisions per axis. The difference of these two strategies is illustrated in Table 7.4.

Table 7.2: Original evaluation grid (1x)

θ/N	952.00	1462.00	1972.00	2482.00	2992.00	3502.00	4012.00	4522.00	5032.00	5542.00	6052.00
3.31	116.64	42.24	58.53	61.26	53.15	47.49	45.86	43.84	39.43	33.10	14.27
10.58	207.96	137.77	145.05	138.78	122.17	108.27	100.50	93.95	84.44	75.92	52.21
17.85	304.47	294.53	271.41	247.90	224.81	203.78	188.44	174.67	156.48	146.42	127.10
25.12	305.95	308.01	312.19	312.78	307.01	294.54	280.81	262.78	235.65	224.02	211.74
32.38	305.95	308.04	315.53	329.98	341.86	344.12	339.23	322.71	291.30	279.03	270.38
39.65	305.95	308.04	315.71	333.29	351.96	362.18	363.65	349.93	318.00	306.32	299.50
46.92	305.95	308.04	315.71	333.87	354.48	367.56	371.75	359.64	328.15	317.26	311.58
54.19	305.95	308.04	315.71	333.97	355.10	369.08	374.25	362.85	331.71	321.34	316.36
61.46	305.95	308.04	315.71	333.98	355.21	369.40	374.81	363.60	332.60	322.41	317.69
68.73	305.95	308.04	315.71	333.98	355.25	369.51	375.01	363.89	332.96	322.87	318.29
75.99	305.95	308.04	315.71	333.99	355.28	369.61	375.22	364.18	333.32	323.33	318.89

Table 7.3: Increased evaluation grid (6x)

θ/N	3.31	10.58	17.85	25.12	32.38	39.65	46.92	54.19	61.46	68.73	75.99
3.31	116.64	44.88	55.04	70.74	49.65	47.57	45.57	45.11	39.38	34.18	14.27
10.58	207.92	140.43	140.79	145.66	117.71	106.39	98.30	95.09	82.63	76.34	50.65
17.85	305.76	303.38	280.64	254.13	226.68	203.92	187.22	178.79	154.20	147.45	126.95
25.12	305.95	307.85	313.50	317.19	313.19	299.14	283.82	272.89	234.84	227.30	214.76
32.38	305.95	307.86	314.68	330.56	346.04	348.28	342.68	334.90	289.87	282.37	273.44
39.65	305.95	307.86	314.71	332.44	354.45	364.75	365.61	361.30	314.74	308.30	301.17
46.92	305.95	307.86	314.71	332.68	356.35	369.38	372.82	370.27	323.75	318.26	312.28
54.19	305.95	307.86	314.71	332.72	356.77	370.61	374.93	373.06	326.74	321.78	316.43
61.46	305.95	307.86	314.71	332.72	356.86	370.94	375.53	373.91	327.70	322.98	317.94
68.73	305.95	307.86	314.71	332.72	356.88	371.02	375.70	374.17	328.00	323.38	318.49
75.99	305.95	307.86	314.71	332.72	356.88	371.04	375.75	374.25	328.10	323.52	318.68

Table 7.4: Lookup table difference between evaluation grid

θ/N	3.31	10.58	17.85	25.12	32.38	39.65	46.92	54.19	61.46	68.73	75.99
3.31	0.00	2.64	-3.49	9.49	-3.50	0.08	-0.29	1.27	-0.05	1.08	0.00
10.58	-0.04	2.67	-4.25	6.88	-4.46	-1.88	-2.20	1.14	-1.81	0.41	-1.56
17.85	1.29	8.85	9.23	6.23	1.87	0.14	-1.22	4.12	-2.28	1.03	-0.15
25.12	0.00	-0.16	1.31	4.41	6.19	4.60	3.01	10.11	-0.81	3.27	3.03
32.38	0.00	-0.18	-0.85	0.58	4.19	4.16	3.45	12.18	-1.43	3.34	3.06
39.65	0.00	-0.18	-1.00	-0.85	2.49	2.57	1.95	11.37	-3.26	1.98	1.67
46.92	0.00	-0.18	-1.01	-1.18	1.87	1.82	1.07	10.62	-4.40	1.01	0.69
54.19	0.00	-0.18	-1.01	-1.25	1.66	1.53	0.68	10.21	-4.98	0.44	0.07
61.46	0.00	-0.18	-1.01	-1.26	1.64	1.54	0.72	10.31	-4.90	0.56	0.25
68.73	0.00	-0.18	-1.01	-1.26	1.63	1.51	0.69	10.28	-4.95	0.51	0.20
75.99	0.00	-0.18	-1.01	-1.27	1.60	1.43	0.53	10.06	-5.22	0.19	-0.21

Table 7.5 shows error statistics of the effect of increasing subdivisions per axis. There is a substantial error reduction by increasing the grid size for table axes. The error reduction is more than 70% by increasing the subdivisions. However, increasing the number of grid points increases the quality of the approximation and minimizes interpolation error, but also increases the computation time.

7.4.2.3 Calibrating the Feature

After the normailser and tables are initilise and filled optimally, the next step is to calibrate the feature as a whole. The entire feature, all the table values are filled by referring to a model. The values of the nomalisers for speed, throttle and time over the range of each variable is initilise and put specified values into

Table 7.5: Difference between evaluation grid

Error Statistics	1x Evaluation Grid	6x Evaluation Grid
Maximum Absolute Error	44.82	12.43
Maximum Square Error	28.27	7.835
Total Square Error	11310	3134

each cell of the two tables. The optimise values are filled in tables by reference to the model.

To populate features with more than one calibration table is more complicated. To generate the calibration one table is filled at a time, the other factors are either held at some mid-range value or specified as a range. If the latter is chosen the table will be filled with calibration that minimises the total error across the range. When a strategy is calibrated by reference to a model, it is useful to compare the strategy and the model. Figure 7.6 compares the model and a full factorial grid filled using the breakpoints.

One of the key areas that can lead to an error is the number of breakpoints within the table, too few will present significant error due to the incapability of representing a complex surface, while too many use too much of ECU memory. Figure 7.6 show the matching error surface between the strategy and the model. The error is more obvious at relative high speed, at a range of 4000–5000 RPM. At this spacing, the placement of the throttle break points at high speed does not match the observed curvature in the sigmoid response characteristic, shown in Figure 7.6(a).

Therefore, the number of breakpoint can be judiciously incremented to satisfy the accuracy requirement across the factor range. The statistics are useful in providing comparisons but they should not be the only metric, the maximum error and goodness of fit across the range should be checked as well as a visual inspection for plausibility. Another feature model, with an incremented break-

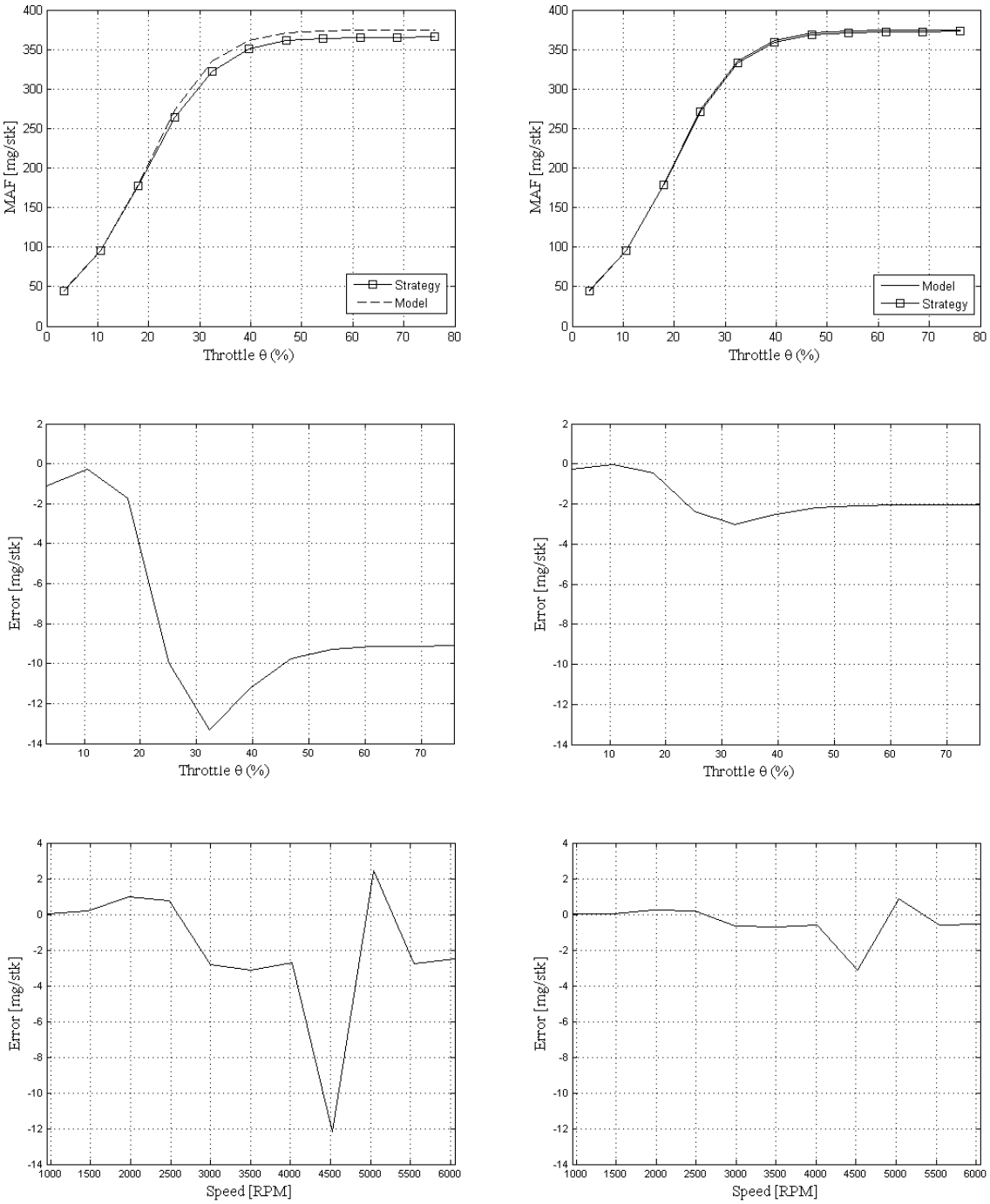


Figure 7.6: Effect of break points on interpolation error at 4500RPM

point is created which significantly reduce the maximum error and improve the fit of the model to the strategy. The visual inspection shows a good calibration fit to the response model with clear reduction in a maximum error and hence a total square error for the surface.

Increasing number of breakpoint reduce the interpolation error, but in the same time increase the memory allocation on the ECU strategy (128), and hence the operationality for the powertrain controller increases. Therefore, optimal lookup table breakpoints should be selected to satisfy only the maximum interpolation error. For the case of transient calibration, there is already an added dimension of time, so any increase in one dimension would exponentially increase the other.

7.5 Applications in the Calibration Process

The feasibility of controlling a 1.4 litre SI engine using full model-based transient calibration process has been demonstrated. This process required very less dynamometer time as compare to the methods used previously by (43; 44; 45), where the complete process took a total of 4 calender months from beginning to end. The process also was able to shift a significant portion of the overall transient calibration process out of the overall transient engine calibration process out of the high cost, high demand transient emissions test cell to the lower cost desktop PC environment, thus freeing up dynamometer time for other engine development process.

The process is scalable and is capable of accommodating highly complex control, including variable valve actuation and aftertreatment systems. Future applications to other engines and engine technologies (such as systems with secondary energy recovery, alternative fuels, or hybrid systems operating on multiple power sources) are also possible. However, adding new control or calibration variable will add required effort to the process, but in this case only in linear fashion and not geometrically as in the case to conventional calibration techniques. With two-stage non-linear model for repeated measurement data, the engine testing

requirement will also not increase significantly more in proportion to the increase in the number of calibrateable parameters. The advantage that the transient calibration process offers is that by using high predictive engine model for calibration, optimisation can occur holistically and efficiently, across the whole cycle simultaneously, rather than in a piece-wise fashion.

Futuremore, there is a wide range of possible optimisation and cost or minimisation functions that can be employed, to allow trade-offs to be investigated between any number of emissions, performance or fuel efficiency measures. Future improvements to the calibration optimisation process will include the development of optimal time-based control schedules. This approach will greatly simplify the development of future control systems for highly complex, multi-parameter engine and aftertreatment control systems.

Future applications of similar model-based techniques may include advanced control of next generation engine and aftertreatment systems.

7.6 Summary

In this chapter the application of the transient engine calibration process has been treated. The fundamental principle was to fit the response model to the feature model in strategy. The main objective was to ensure that the calibrated strategy and the model were in closest possible agreement.

To evaluate the feature side by side with the model, a strategy was created that takes all of the same variables as the model. Hence, a simple strategy to calibrate a feature for transient MAF adds a 2-D table of speed vs. throttle to the 1-D table that account for the behaviour of the model as the time varies. The transient model for the throttle inferred airflow was fitted to the lookup tables in the strategy. The process has been illustrated for the employment of the developed model for calibration studies. Also, the method to control maximum interpolation errors through the selection of an appropriate table index scheme

and associated table values was discussed.

The new strategy has shown improvement by optimising the breakpoints, with a reduction in the maximum interpolation error. The same effect is noticed by increasing the size of the evaluation grid, and adding more breakpoints to lookup table. For the case of transient calibration, there is already an added dimension of time, so any increase in one dimension would exponentially increase the other, and will affect the overall memory allocation in ECU. Therefore, optimal lookup table breakpoints selection, and judiciously increment in the number of breakpoints have an convincing effect in the transient calibration optimisation problem.

8

Conclusion And Future Work

8.1 Conclusion

The thesis has detailed the concept of utilising a model based calibration approach applied to the transient engine calibration. The main goal of the modelling was to predict the responses of the interest in a time domain, in terms of the primary engine state and actuation variables over the entire region of operability.

The research has build on the application of hierarchical statistical modelling methods, and analysis of repeated experiments for the application of engine mapping. The methodology is based on two-stage regression approach, which organise the engine data for the mapping process in *sweeps*.

Previous work concentrated on the development of model based calibration techniques that was only limited to steady stage condition. The models were based on the assumption that the covariate vector at the global model summarising the individual characteristic is constant across the observation, the value of the regression parameter for individual at local model remain fixed for that individual over the course of observation. This assumption is particularly not true in engine transient phenomenon such as engine warm-up and fuel dynamic response characteristics, where the individual specific information may change during the

course of observation, to exhibit corresponding changes at different time.

The models were extended here with the modification of the two-stage regression methodology with the introduction of *time-dependent covariates* in the hierarchy of stage-1 modelling. Sophisticated hierarchical empirical model capable of accurately predicting engine response characteristics over the entire region of operability lie at the heart of the methodology. The following section highlight the contribution to the research in this thesis;

8.1.1 Two-Stage Regression

For the steady state mapping, a single experimental design was specified to characterise the primary and secondary feature models. The primary feature was taken as MBT spark, which was identified by modelling torque response using MLP neural network at the second stage of the model. The secondary models were exhaust gas temperature and residual fraction to show the quality of the response feature models generated. It has been shown that these models exhibit satisfactory statistics to confirm the applicability of the approach, and feasibility of MLP network in these models.

- The space filling design allows the flexibility to generate response models ranging from low order polynomials to advanced models such as neural network. Also, the space filling design do not depend on model type; and the most suitable model can be choose to construct a design, and when data is collected, a different model type can be tried that produces the best. It is of convenience when different models are to be compared with same set of data.

- Mulitlayer perceptron (MLP), a neural network based model was used for generation of response models in the multi-covariate case. The neural network based models represent a class of non-parametric estimator which allow complex surfaces to be realised over the entire region of operability with accurate response feature predictions. A comparison between the MLP based neural network and radial basis functions (RBF) model fits to

the data suggest that MLP offers marginally improved performance over their RBF counterparts. The basis of this comparison was made on assessing relevant model selection criteria, as well as internal and external validation fits.

- The number of parameter fits in the MLP neural network compare to other modelling methods increases. Therefore, for the approximation of a nonlinear input-output mapping, the MLP require a smaller number of parameters then the RBF network for the same degree of accuracy. The ability of MLP networks to fit a variety of response characteristics collected as the main advantage of these models.

8.1.2 Transient Engine Model

The two-stage regression approach was extended with a slight modification to the general structure at stage-1 in the hierarchy to permit the individual regression parameters, to depend on changing individual-specific information while handling time-varying individual attributes. For the transient engine model, the empirical regression was demonstrated through its application to a practical example. For this purpose,

- A throttle body inferred transient air flow phenomenon was addressed. The population of throttle angle θ and engine speed N look-up table at the transient state condition is considered.

- The data collection for the transient engine model was performed on a transient test bed, with sophisticated software and hardware installed on it. In all cases engine testing has been performed using automated control of the dynamometer. A custom made NI LabView software was designed and build to allow transient testing of the powertrain under repeatable conditions in a laboratory environments, and to allow a comprehensive control of the dynamometer controller behaviour. An ATI VISION software coupled with a VISION hub was used to control and measure the data from the ECU and external sources of the engine.

- In the development of the local model, the Richard growth model was selected. It has been shown that the Richard growth model has the ability to address both the sigmoid as well as concave nature of the curve which was obtained at different level of speed. The growth model was defined in a Simulink and Matlab environment, for its used in MBC environments for the analysis purpose.

- The MLP network was fitted to the second stage of the model where it outperform the other models of the choice (*i.e.,* Polynomial and RBF), where the later lack some of the critical curves in the response features, and was demonstrated not suitable for mass air flow prediction. Neural network model has shown good fit in external validation process.

8.1.3 Application of Transient Engine Model in Calibration

Finally, the general ability of the model was demonstrated in CAGE through the course of the work to the calibration problem. The transient model for the throttle inferred airflow was used in generation of the lookup tables in the strategy. The function model in toolbox permits alignment of the strategy and reference model inputs, which is a fundamental requirement for lookup-table population.

- The calibration generation is heavily dependent upon the reference model predictions, and any error in the model will be reflected in the final lookup tables. Therefore, emphasis should be on building a transient engine model with accurate fit to the data and having good prediction capability.

- Optimal lookup table breakpoints should be selected to satisfy only the maximum interpolation error, that are capable of representing a complex surface and decrease the memory allocation on the ECU strategy. This is very important for transient engine calibration generation where there is a added dimension of time, and any increase in one dimension would exponentially increase the other.

8.2 Future Work

The work is concentrated on the analysis of a small transient engine data set, permitting the consideration of relatively simple model for engine calibration. However, the work can be extended to a fully functional methodology, which covers the entire engine operating range and optimise for different set of engine calibration requirements.

Different prototype of engine in the development stage can be compared in a meaningful way in order to access the effect of design changes. To compare engines at a predefined set of observational points or at a set of values defined without reference to exhaust emissions does not revel the whole story, because in practice the engines will be operated, not at these values, but at their optimum calibrations. It is impractical to compare engines in this way if it takes an excessive amount of time to establish a near optimal calibration. A calibration process can be established to use them to compare different engines, and it could be embedded experimental design to investigate the effect of design changes. Engine could then be used to compare the total amount of fuel consumed and its emission output over any specific drive cycle using the calibration optimal for that engine in a specified vehicle.

The aim was to introduce a transient engine model for calibration based on a two-stage analysis of the data. It has been shown that these method are simple to construct and provide numerous advantages over a large polynomial model. These model are easier to interpret for their inadequacies which lead to their improvement.

References

[1] K. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. CRC Press, 2006. vii, 21, 35, 42, 48

[2] DW Rose, M. Cary, SB Zulczyk, R. Sbaschnig, and KM Ebrahimi. C606/025/2002 An engine mapping case study-a two-stage regression approach. In *International Conference on Statistics and Analytical Methods in Automotive Engineering*, page 53. Wiley, 2002. viii, 18, 85, 92, 94

[3] P.J. Maloney. Objective Determination of Minimum Engine Mapping Requirements for Optimal SI DIVCP Engine Calibration. *SAE Technical Paper Series No. 2009-01-0246*, 2009. viii, 74, 79, 100, 103

[4] J. Heywood. Internal combustion engine: fundamentals (ISE). 1989. ix, 118, 119, 120, 136

[5] H. Stuhler, K. Gschweitl, T. Kruse, A. Stuber, W. Piock, H. Pfluegl, and P. Lick. Automated model-based GDI engine calibration adaptive online DoE approach. *SAE Technical Paper Series No. 2002-01-0708*, 2002. 2

[6] Y. Nozaki, T. Fukuma, and K. Tanaka. Development of a rule-based calibration method for diesel engines. *SAE Technical Paper Series No. 2005-01-0044*, 2005. 2

[7] E. Rask and M. Sellnau. Simulation-based engine calibration: Tools, techniques, and applications. *SAE SP*, pages 21–32, 2004. 2

[8] K. Suzuki, M. Nemoto, and K. Machida. Model-based calibration process for producing optimal spark advance in a gasoline engine equipped with a variable valve train. *SAE Paper*, pages 01–3235, 2006. 2

[9] G.E.P. Box and N.R. Draper. *Empirical model-building and response surface*. 1986. 2

[10] R.H. Myers and D.C. Montgomery. *Response surface methodology*. John Wiley & Sons Inc, 2005. 2

[11] D.T. Montgomery and R.D. Reitz. Optimization of heavy-duty diesel engine operating parameters using a response surface method. *SAE Technical Paper Series No. 2000-01-1962*, 2000. 2

[12] D.T. Montgomery and R.D. Reitz. Effects of multiple injections and flexible control of boost and EGR on emissions and fuel consumption of a heavy-duty diesel engine. *SAE Technical Paper Series No. 2001-01-0195*, 2001. 2, 23

[13] S.P. Edwards, A.D. Pilley, MS Michon, and MG Fournier. The Optimization of Common Rail Fie Equipped Engines Through the Use of Statistical Experimental Design, Mathematical Modeling and Genetic Algorithms. *SAE Technical Paper Series No. 970346*, 1997. 2

[14] R. Burk, R. Wakeman, and F. Jacquelin. A contribution to predictive engine calibration based on vehicle drive cycle performance. *SAE Technical Paper Series No. 2003-01-0225*, 2003. 2

[15] P. Dimopoulos, E. Vaccarino, A. Schoni, C. Operti, A. Eggegglmann, and C. Sparti. Statis-

tical methods for solving the fuel consumption/emission conflict on DI diesel engines. *SAE Technical Paper Series No. 1999-01-1077*, 1999. 2

[16] Paulitsch R. Kampelmuhler, F.T. and Gschweitl. Automatic ECU-Calibration An Alternative to Conventional Methods. *SAE Technical Paper Series No. 930395*, 1993. 2

[17] Oligschlager. U. Eifler. G. Schmitz., G. and H. Lechner. Automated System for Optimized Calibration of Engine Management Systems. *SAE Technical Paper Series No. 940151*, 1994. 2

[18] T. Brooks, G. Lumsden, and H. Blaxill. Improving base engine calibrations for diesel vehicles through the use of DoE and optimization techniques. *SAE Technical Paper Series No. 2005-01-3833*, 2005. 2

[19] A. Knafl, J.R. Hagena, Z. Filipi, and D. Assanis. Dual-Use Engine Calibration: Leveraging Modern Technologies to Improve Performance-Emissions Tradeoff. *SAE Technical Paper Series No. 2005-01-1549*, 2005. 2

[20] L. Hernandez, J.M. Desantes, J.J. Lopez, and J.V. Garcia. Application of neural networks for prediction and optimization of exhaust emissions in a HD diesel engine. *SAE Technical Paper Series No. 2002-01-1144*, 2002. 2

[21] I. Brahma and C.J. Rutland. Optimization of diesel engine operating parameters using neural networks. *SAE transactions*, 112(4):2521–2529, 2003. 2

[22] Y. He and C.J. Rutland. Modelling of a turbocharged DI diesel engine using artificial neural networks. *SAE Technical Paper Series No. 2002-01-2772*, 2002. 2

[23] Y. He and CJ Rutland. Application of artificial neural networks in engine modelling. *International Journal of Engine Research*, 5(4):281–296, 2004. 2

[24] I. Brahma and C.J. Rutland. Improvement of neural network accuracy for engine simulations. *SAE Technical Paper Series No. 2003-01-3227*, 2003. 2

[25] R. Mller and B. Schneider. Approximation and Control of Engine Torque Using Neural Networks. *SAE Technical Paper Series No. 2000-01-0929*, 2000. 2

[26] R. Mueller and H.H. Hemberger. Neural adaptive ignition control. *SAE Technical Paper Series No. 981057*, 1998. 2

[27] T.S. Rognvaldsson, C. Carlsson, M. Hellring, M. Larsson, T. Munther, and N. Wickstrom. Spark Advance Control Using the Ion Current and Neural Soft Sensors. *SAE Technical Paper Series No. 1999-01-1162*, 1999. 2

[28] C.N. Grimaldi and F. Mariani. OBD engine fault detection using a neural approach. *SAE Technical Paper Series No. 2001-01-0559*, 2001. 2

[29] M.L. Traver, R.J. Atkinson, and C.M. Atkinson. Neural network-based diesel engine emissions prediction using in-cylinder combustion pressure. *SAE transactions*, 108(4):1166–1180, 1999. 2

[30] C.M. Atkinson, T.W. Long, and E.L. Hanzevack. Virtual sensing: a neural network-based intelligent performance and emissions prediction system for on-board diagnostics and engine control. *PROGRESS IN TECHNOLOGY*, 73:301–314, 1998. 2, 4, 115, 116

[31] M. Ayeb, D. Lfichtenthaler, T. Winsel, and HJ Theuerkauf. SI engine modeling using neural networks. *SAE SPEC PUBL, SAE, WARRENDALE, PA,(USA), Feb 1998,*, 1357:107–115, 1998. 2

[32] D. Lichtenthaler, M. Ayeb, H.J. Theuerkauf, and T. Winsel. Improving real-time SI engine models by integration of neural approximators. *SAE Technical Paper Series No. 1999-01-1164*, 1999. 2

[33] R. Hentschel, R. Cernat, and J.U. Varchmin. In-Car Modeling of Emissions with Dynamic Artificial Neural Networks. *SAE Technical Paper Series No. 2001-01-3383*, 2001. 2

[34] I. Brahma, MC Sharp, IB Richter, and TR Frazier. Development of the nearest neighbour multivariate localized regression modelling technique for steady state engine calibration and comparison with neural networks and global regression. *International Journal of Engine Research*, 9(4):297–323, 2008. 2

[35] F. Yang, J. Zhang, O. Minggao, and H. Qiang. Study on Modelling Method for Common-Rail Diesel Engine Calibration and Optimization. *SAE Technical Paper Series No. 2004-01-0426*, 2004. 2

[36] M. Guerrier and P. Cawsey. The development of model based methodologies for gasoline IC engine calibration. *SAE transactions*, 113(3):981–1002, 2004. 3, 18, 27, 84, 146

[37] J. Bayer and D.E. Foster. Zero-dimensional soot modeling. *SAE transactions*, 112(3):1446–1458, 2003. 3

[38] I. Brahma, C.J. Rutland, D.E. Foster, and Y. He. A New Approach to System Level Soot Modeling. *SAE Technical Paper Series No. 2005-01-1122*, 2005. 3

[39] D. Mowll and DR Robinson. Bayesian experimental design and its application to engine research and development. Technical report, Society of Automotive Engineers, 400 Commonwealth Dr, Warrendale, PA, 15096, USA,, 1996. 3

[40] A.D. Pilley, D. Mowll, and DR Robinson. Optimizing Engine Performance and Emissions Using Bayesian Techniques. *SAE Technical Paper Series No. 971612*, 1997. 3

[41] Ricardo Inc. Ricardo consulting. `http://www.ricado.com/`, October 2010. 3

[42] C.M. Atkinson, M. Allain, and H. Zhang. Using Model-Based Rapid Transient Calibration to Reduce Fuel Consumption and Emissions in Diesel Engines. *SAE Technical Paper Series No. 2008-01-1365*, 2008. 3, 116

[43] C. Atkinson. Dynamic Model-Based Calibration Optimization: An Introduction and Application to Diesel Engines. *SAE Technical Paper Series No. 2005-01-0026.* 3, 13, 21, 146, 160

[44] C.M. Atkinson, M. Allain, Y. Kalish, and H. Zhang. Model-Based Control of Diesel Engines for Fuel Efficiency Optimization. *SAE Technical Paper Series No. 2009-01-0727.* 3, 116, 146, 160

[45] I. Brahma, M.C. Sharp, and T.R. Frazier. Empirical modeling of transient emissions and transient response for transient optimization. *SAE International Journal of Engines*, 2(1):1433, 2009. 3, 116, 160

[46] M. Hafner. Model-Based Determination of Dynamic Engine Control Function Parameters. *SAE Technical Paper Series No. 2001-01-1981*, 2001. 3, 116

[47] H.P. Dohmen and G. Fehl. Dynamic and Transient Engine Testing Application, System Requirements and Modular Structures. *SAE Technical Paper Series No. 982958*, 1998. 3

[48] SP Stevens, PJ Shayler, and TH Ma. Experimental data processing techniques to map the performance of a spark ignition engine. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 209(4):297–306, 1995. 11, 14, 24, 62, 63, 78, 95

[49] T. Holliday, A.J. Lawrance, and T.P. Davis. Engine-mapping experiments: a two-stage regression approach. *Technometrics*, pages 120–126, 1998. 11, 14, 16, 62, 63, 75, 95, 97, 99

[50] T. Holliday. *The Design and Analysis of Engine Mapping Experiments: A Two-stage Approach.* University of Birmingham, 1995. 11, 14, 16, 63, 95, 99

[51] TP Davis and AJ Lawrance. Engine Mapping: a two-stage regression approach based on spark sweeps. In *Statistic of Engine Optimization*, pages 99–108, 2000. 11

[52] TD Barker. Engine mapping techniques. *International Journal of Vehicle Design*, 3:142–52, 1982. 13, 14, 16, 63

[53] L.A.D. Sheridan, R. Goyder, J.B. Cherrie, and T.M. Morton. Defining a model-based calibration process for a twin-independent valve timing engine. In *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004*, volume 2, 2004. 13

[54] K. R
"opke and C. von Essen. DoE in engine development. *Quality and Reliability Engineering International*, 24(6):643–651, 2008. 13

[55] M. Davidian and D.M. Giltinan. *Nonlinear models for repeated measurement data.* Chapman & Hall/CRC, 1995. 16, 64, 117

[56] M.J. Crowder and DJ Hand. *Analysis of repeated measures.* Chapman & Hall/CRC, 1990. 16, 67, 96, 101

[57] RE Baker and EE Daby. Engine mapping methodology. *SAE paper*, 770077, 1977. 16, 63

[58] M. Cary. *A Model Based Engine Calibration Methodology for a Port Fuel Injection, Spark-Ignition Engine.* University of Bradford, 2003. 18, 62, 67, 78, 82, 84, 95, 99, 103, 110, 111, 130, 148, 154, 155

[59] C. Tindle. Cold Engine Emissions Optimization Using Model Based Calibration. 18

[60] T.W. Simpson, JD Poplinski, P.N. Koch, and J.K. Allen. Metamodels for computer-based engineering design: survey and recommendations. *Engineering with Computers*, 17(2):129–150, 2001. 21, 22, 32, 60

[61] G.G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129:370, 2007. 21, 32

[62] G.E.P. Box and KB Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45, 1951. 23

[63] G. Taguchi et al. Design of experiments. *Maruzen Co., Ltd*, 1976. 23

[64] A. Dey and R. Mukerjee. *Fractional factorial plans.* Wiley New York, 1999. 23, 26

[65] D.C. Montgomery. *Design and analysis of experiments.* John Wiley & Sons Inc, 2008. 23, 26, 29

[66] W.J. Diamond. *Practical experiment designs for engineers and scientists*. John Wiley & Sons Inc, 2001. 23

[67] PJ Shayler, PR Tinwell, J. Dixon, and D. Eade. A Development Methodology for Improving the Cold Start Performance of Spark Ignition Engines. *SAE Technical Paper Series No. 940084*, 1994. 23

[68] MC Bates and M. Heikal. A knowledge-based model for multi-valve diesel engine inlet port design. *SAE Technical Paper Series No. 2002-01-1747*, 2002. 23

[69] J. Seabrook. Practical implementation of design of experiments in engine development. *Statistics for Engine Optimization*, page 145, 2000. 23

[70] A. Ghauri, S.H. Richardson, and C.J.E. Nightingale. Variation of Both Symmetric and Asymmetric Valve Events on a 4-Valve SI Engine and the Effects on Emissions and Fuel Economy. *SAE Technical Paper Series No. 2000-01-1222*, 2000. 24

[71] S. Flint and P. Cawsey. Use of Experimental Design and Two Stage Modeling in Calibration Generation for Variable Camshaft Timing Engines. *Design of Experiments (DOE) in der Motorenentwicklung, Expert Verlag, ISBN*, pages 3–8169. 24

[72] C. Croarkin, P. Tobias, JJ Filliben, B. Hembree, WF Guthrie, J. Prins, C. Zey, NA Heckert, and L. Trutna. NIST/SEMATECH e-handbook of statistical methods, 2002. 24

[73] R.H. Myers, D.C. Montgomery, and C.M. Anderson-Cook. Response surface methodology. 1995. 25

[74] JM Lucas. Using response surface methodology to achieve a robust process. *Journal of Quality Technology*, 26(4):248–260, 1994. 27

[75] J. Seabrook, B. ROGERS, G. FARROW, and J. PATTERSON. Applications of advanced modelling methods in engine development. In *International Conference on Statistics and Analytical Methods in Automotive Engineering*, page 17. Wiley, 2002. 30

[76] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989. 30, 32, 33, 40, 79

[77] R. Jin, W. Chen, and T.W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, 2001. 31

[78] JR Koehler and AB Owen. Computer experiments. *Handbook of statistics*, 13:261–308, 1996. 31

[79] MD McKay, RJ Beckman, and WJ Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, pages 55–61, 2000. 33, 79

[80] B. Tang. Orthogonal array-based Latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397, 1993. 36

[81] LM Johnson et al. Minimax and maximin distance designs* 1. *Journal of statistical planning and inference*, 26(2):131–148, 1990. 36

[82] V. Myers. Response surface methodology. *Technometrics*, 44(3):298–299, 2002. 38

[83] M.D. Morris and T.J. Mitchell. Exploratory designs for computational experiments* 1. *Journal of Statistical Planning and Inference*, 43(3):381–402, 1995. 38

[84] T.W. Simpson, T.M. Mauery, J.J. Korte, and F. Mistree. Comparison of response surface and kriging models for multidisciplinary design optimization. *AIAA paper 98*, 4758(7). 40

[85] W.J. Welch, J. Sacks, H.P. Wynn, T.J. Mitchell, and M.D. Morris. Screening, predicting, and computer experiments. *Technometrics*, 34(1):15–25, 1992. 40

[86] M. Meckesheimer, R.R. Barton, T. Simpson, F. Limayem, and B. Yannou. Metamodeling of combined discrete/continuous responses. *AIAA journal*, 39(10):1950–1959, 2001. 41

[87] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991. 41

[88] G. Wahba. Spline models for observational data, vol. 59. In *SIAM. CBMS-NSF Regional Conference Series in Applied Mathematics*, 1990. 41

[89] C.J. Stone, M.H. Hansen, C. Kooperberg, and Y.K. Truong. Polynomial splines and their tensor products in extended linear modeling. *The Annals of Statistics*, 25(4):1371–1425, 1997. 41

[90] P.H.C. Eilers and B.D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121, 1996. 41

[91] D. Ruppert and R.J. Carroll. Spatially-adaptive Penalties for Spline Fitting. *Australian & New Zealand Journal of Statistics*, 42(2):205–223, 2000. 41

[92] J.H. Friedman. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991. 42

[93] A. Sudjianto, L. Juneja, H. Agrawal, and M. Vora. Computer aided Reliability and robustness assessment. 43

[94] B. Evans and D. Fisher. Overcoming process delays with decision tree induction. *IEEE Expert*, 9(1):60–66, 2002. 44

[95] M.T. Hagan, H.B. Demuth, and M. Beale. *Neural network design*. PWS Publishing Co. Boston, MA, USA, 1997. 45

[96] Y. He and C.J. Rutland. Modelling of a turbocharged DI diesel engine using artificial neural networks. *SAE Technical Paper Series No. 2002-01-2772*, 2002. 46

[97] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, 1999. 49, 52, 56, 59, 103

[98] Y. Le Cun. Efficient learning and second order methods. In *Tutorial presented at Neural Information Processing Systems*, volume 5, 1993. 49

[99] PJ Shayler, PI Dow, DJ Hayden, and G. Horn. Using neural networks in the characterization and manipulation of engine data. *Statistics for Engine Optimization*, page 145, 2000. 50

[100] A.P. Engelbrecht. *Computational intelligence: An introduction*. Wiley, 2007. 50

[101] G. Thimm, E. Fiesler, and M. IDIAP. High-order and multilayer perceptron initialization. *IEEE Transactions on Neural Networks*, 8(2):349–359, 1997. 50

[102] LFA Wessels, E. Barnard, and P. CSIR. Avoiding false local minima by proper initialization of connections. *IEEE Transactions on Neural Networks*, 3(6):899–905, 1992. 50

[103] JT Tou and RC Gonzalez. Pattern recognition principles. 1974. *Addision Wesley, Massachusetts*. 51

[104] R.P. Lippmann. An introduction to computing with neural nets. *ARIEL*, 209:115–245. 56

[105] E. Levin, N. Tishby, and S. Solla. A statistical approach to learning and generalization in layered neural networks. *Proceedings of the IEEE*, 78(10):1568–1574, 1990. 56

[106] Y. Bengio. *Neural networks for speech and sequence recognition*. Van Nostrand Reinhold, 1995. 57

[107] V.C.P. Chen, K.L. Tsui, R.R. Barton, and M. Meckesheimer. A review on design, modeling and applications of computer experiments. *IIE transactions*, 38(4):273–291, 2006. 60

[108] M.J. Lindstrom and D.M. Bates. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 46(3):673–687, 1990. 61

[109] Z. Mencik, PN Blumberg, and Society of Automotive Engineers. Representation of Engine Data by Multi-Variate Least-Squares Regression. Technical report, Society of Automotive Engineers, 400 Commonwealth Dr, Warrendale, PA, 15096, USA,, 1978. 63

[110] BK Powell and JA Cook. Nonlinear low frequency phenomenological engine modeling and analysis. In *American Control Conference, 1987*, pages 332–340. IEEE, 2009. 63

[111] PR Crossley and JA Cook. A nonlinear engine model for drivetrain system development. In *Control 1991. Control'91., International Conference on*, pages 921–925. IET, 2002. 63

[112] M. Davidian and D.M. Giltinan. Analysis of repeated measurement data using the nonlinear mixed effects model. *Chemometrics and Intelligent Laboratory Systems*, 20(1):1–24, 1993. 64

[113] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence*, volume 14, pages 1137–1145. Citeseer, 1995. 69

[114] M.J.L. Orr. *Introduction to Radial Basis Function Networks,*. Technical Report, Centre for Cognitive Science, University of Edinburgh, 1996. 69, 70

[115] G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979. 70

[116] B.D. Ripley. Statistical ideas for selecting network architectures. *Neural Networks: Artificial Intelligence and Industrial Applications*, pages 183–190, 1995. 70

[117] N. Sugiura. Further analysts of the data by Akaike's information criterion and the finite corrections. *Communications in Statistics-Theory and Methods*, 7(1):13–26, 1978. 70

[118] K.P. Burnham and D.R. Anderson. Multimodel inference: understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2):261, 2004. 70

[119] K. Morita, Y. Sonoda, T. Kawase, and H. Suzuki. Emission Reduction of a Stoichiometric Gasoline Direct Injection Engine. *SAE Technical Paper Series No. 2005-01-3687*, 2005. 74

[120] GEP Box and DW Behnken. Some new three level designs for the study of quantitative variables. *Technometrics*, pages 455–475, 1960. 78

[121] Model-Based Calibration Toolbox. 84, 88, 148

[122] D.J.C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992. 103

[123] G.A.F. Seber and C.J. Wild. *Nonlinear regression*. Wiley-IEEE, 2003. 108, 117, 130

[124] FJ Richards. A flexible growth function for empirical use. *J. exp. Bot*, 10(29):290–300, 1959. 130, 136

[125] YC Lei and SY Zhang. Features and partial derivatives of Bertalanffy-Richards growth model in forestry. *Nonlinear Analysis: Modelling and Control*, 9(1):65–73, 2004. 130, 131

[126] D. Fekedulegn, M.P. Mac Siurtain, and J.J. Colbert. Parameter estimation of nonlinear growth models in forestry. *Silva Fennica*, 33(4):327–336, 1999. 131

[127] S. Chen, ES Chng, and K. Alkadhimi. Regularized orthogonal least squares algorithm for constructing radial basis function networks. *International Journal of Control*, 64(5):829–837, 1996. 137

[128] RJ Lygoe. Fitting automotive microprocessor control look-up tables to a response surface model using optimization methods. *SAE Technical Paper Series No. 981459*, 1998. 147, 151, 153, 160