APPLICATION OF MULTIDISCIPLINARY DESIGN OPTIMISATION TO ENGINE CALIBRATION OPTIMISATION

XUEFEI YIN

Submitted for the degree

of Doctor of Philosophy

School of Engineering, Design and Technology

University of Bradford

2012

Abstract

Automotive engines are becoming increasingly technically complex and associated legal emissions standards more restrictive, making the task of identifying optimum actuator settings to use significantly more difficult. Given these challenges, this research aims to develop a process for engine calibration optimisation by exploiting advanced mathematical methods. Validation of this work is based upon a case study describing a steady-state Diesel engine calibration problem.

The calibration optimisation problem seeks an optimal combination of actuator settings that minimises fuel consumption, while simultaneously meeting or exceeding the legal emissions constraints over a specified drive cycle. As another engineering target, the engine control maps are required as smooth as possible.

The Multidisciplinary Design Optimisation (MDO) Frameworks have been studied to develop the optimisation process for the steady state Diesel engine calibration optimisation problem. Two MDO strategies are proposed for formulating and addressing this optimisation problem, which are All At Once (AAO), Collaborative Optimisation. An innovative MDO formulation has been developed based on the Collaborative Optimisation application for Diesel engine calibration.

Form the MDO implementations, the fuel consumption have been significantly improved, while keep the emission at same level compare with the bench mark solution provided by sponsoring company. More importantly, this research has shown the ability of MDO methodologies that manage and organize the Diesel engine calibration optimisation problem more effectively.

Acknowledgements

It would not have been possible to undertake and write this doctoral thesis without the help and support of many kind people around me, only some of whom it is possible to give particular mention to here.

First of all I owe my deepest gratitude to my principal supervisor Dr Felician Campean for his supervision, encouragement and support. Felician was always there to support not only my academic work but also my life. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. I also thank my co-supervisor, Prof Alastair Wood, for his academic support that directed me to such a wonderful treasure of this research area, for which I am extremely grateful. This work would not have been possible without Dr Mark Goodman's support from the sponsoring company (Jaguar Land Rover). His help was always prompt and kept my research relevant.

I would also like to make a special reference to my ex-supervisor, Dr Bill Seale. I am honoured to have received his support through his strong internal combustion engine knowledge. I would like to thank Prof Andrew Day and Dr Jian-Ping Li, who did proofreading of my thesis, for which I am really grateful.

I have to thank my best mate from the University of Bradford, Sam Brook, who has been a soul mate. He has been the one who could understand me the most and does not speak any Chinese. He has given the important encouragement to me all the time not only for my research, also my cooking skill.

I would also like to acknowledge and thank the School of Engineering, Design and Technology, at the University of Bradford, and its staff, and the sponsoring company, particularly for the award of a postgraduate research studentship that provided necessary financial support for this research.

Last, but not least, I thank my parents and family for their unconditional support and encouragement throughout, and for believing in me, as always, for which my mere expression of thanks does not suffice.

Table of Contents

Abstract I

Acknowledgements II

Table of Contents III

Nomenclature IX

Chapter 1 Introduction 1

1.1 Engine Mapping and Calibration 1

1.2 Project Background 2

1.3 Aim and Objectives 4

1.4 Methodology 5

1.5 Outline of Thesis 6

Chapter 2 Literature Review: Optimisation Methods and Frameworks 8

2.1 Introduction 8

2.2 Optimisation Problem 8

2.2.1 Gradient-based Algorithms 9

2.2.2 Population-based Algorithms 11

2.3 Multidisciplinary Design Optimisation (MDO) 24

2.3.1 Single-level MDO Strategies 26

2.3.2 Multi-level strategies 32

2.4 Discussion and Conclusion 41

2.5 Summary 42

Chapter 3 Literature Review: Diesel Engine Mapping and Calibration 43
3.1 Introduction 43
3.2. Overview of Diesel Engine Technology 43
3.2.1 Emissions control for Diesel engines 43
3.2.2 Turbo-charging Systems for Diesel Engines 47
3.2.3. Exhaust Gas Recirculation System 48
3.2.4 Common rail injection system 49
3.2.5 Control Strategies for Diesel Engines 51
3.3. Model Based Calibration Process 54
3.3.1 Engine Control Unit Strategy 54
3.3.2 Overview of Model Based Calibration Process 55
3.4. Engine Statistical Modelling 60
3.4.1. Introduction 60
3.4.2 Overview of DoE for Engine Testing 60
3.4.3. Overview of Engine Response Modelling 64
3.5 Review of Diesel Engine Calibration 68
3.6 Summary 75
Chapter 4 Critical Review of Current Process for Diesel Engine Steady State Calibration Process 79
4.1 Introduction 79
4.2 Critical Review of the Current Process 80
4.2.1 Selection of Engine Testing Points 81

4.2.2 Local DoE Investigation at Each Minimap Point 84

4.2.3 Fitting Local Engine Response Model 89

4.2.4 Local and Global Optimisation 96

4.2.5 Validated Calibration Development Process 106

4.3 Critique of the Current Optimisation Process 107

4.4 Calibration Optimisation Problem Re-Analysed 110

4.5 Summary 115

Chapter 5 "All At Once" Multi-disciplinary Design Optimisation Approach to Diesel Calibration Optimisation 117

5.1 Introduction: 117

5.2 Diesel Engine Calibration Problem Analysis and Formulation as MDO / AAO 118

5.3 Algorithm Development and Implementation in Matlab 122

5.3.1. Development of Data Structures for AAO Implementation 123

5.3.2. MATLAB Implementation 126

5.4 Preliminary Results and Analysis 129

5.4.1 AAO Results for Three Minimap Points 129

5.4.2 Study of Effect of Population Size on Algorithm Performance 130

5.4.3. AAO Results Analysis 132

5.5 Preliminary AAO Implementation over the Whole Drive Cycle: 136

5.6 Constraints Analysis: 138

5.6.1 Methodology for Constraints Analysis 138

5.6.2 Results and Analysis: 139

5.7 Development of an Improved GA Creation Function 146

5.8 AAO Results with Corrected GA Creation Function 148

5.8.1. Initial Results and Analysis 148

5.8.2. Further Study of the GA Algorithm Performance 150

5.8.3 Development of Revised GA Operators 151

5.8.4. AAO Implementation with Modified GA Operators and Results Discussion 153

5.7 Summary 159

Chapter 6 Collaborative Optimisation Approach to Diesel Calibration Optimisation 161

6.1 Introduction 161

6.2 Collaborative Optimisation Formulation for Diesel Engine Calibration 162

6.2.1 TCO Formulation for a Diesel Engine Calibration 164

6.2.2 CO Implementation in MATLAB for a Diesel Engine Calibration 168

6.3 TCO Results and Analysis: 170

6.3.1 Three minimap Points Results Analysis: 171

6.3.2 Full Cycle Results Analysis: 173

6.5 Discussions and Conclusions: 179

Chapter 7 Further Development of Collaborative Optimisation for Diesel Engine Calibration 181

7.1 Introduction 181

7.2 Diesel Calibration Problem: Review and Analysis 182

7.2.1 Review and Analysis of Problem Difficulty 182

7.2.2 Optimisation Problem Re-formulation 183

7.3 Formulation and Implementation 185

7.3.1 Mathematical Formulation of “Smoothness” 185

7.3.2 HCO Mathematical Formulation 188

7.3.3 MATLAB Implementation 190

7.4 Tests and Results 192

7.4.1 Three minimap point test... 192

7.4.2 Tests and results for 10 minimap points... 198

7.5 Full Cycle HCO Results Performance Analysis: 206

7.6 Summary 208

Chapter 8 Discussions 209

8.1 Scope of research... 209

8.2 Critical Review of Methodology and Developments...................................... 210

8.3 Discussion of Diesel Engine Calibration Optimisation Problem 211

8.3.1 Discussion of MDO/AAO implementation... 213

8.3.2 Discussion of MDO/CO implementation... 214

8.3.3 Discussion of MDO/HCO Implementation 215

8.4 Discussions of MDO implementation and results.. 217

8.5 Limitations of this research 220

Chapter 9 Conclusions and Recommendations for Further Work 222

9.1 Conclusions 222

9.2 Summary of Contribution: 225

9.3 Recommendations for Future Work 226

Reference 228

Nomenclature

AAO	All At Once framework
ATC	Analytical Targets Cascading framework
BLISS	Bi-Level Integrated System Synthesis framework
BTDC	Before Top Dead Centre
CO	Carbon monoxide
DoE	Design of Experiments
EM-Lims	Emission Limits
EMC	Engine Mapping Calibration
FTP	Federal Test Procedure
GA	Genetic Algorithm
GC	Actuator Gradient Change
HC	Hydrocarbon
HCO	Hybrid Collaborative Optimisation
IDF	Individual Disciplinary Feasible
IMEP	Indicated Mean Effective Pressure
LB,UB	Lower Bound, Upper Bound

MBC	Model Based Calibration
MDF	Multiple Disciplinary Feasible
MDO	Multidisciplinary Design Optimisation
NBI	Normal Boundary Intercept
NEDC	New European Drive Cycle
NOx	Mono-nitrogen oxides
OLH	Optimal Latin Hypercube
PM	Particulates Matter
PRESS	Prediction Error Sum Squares
PSO	Particle Swarm Optimisation
RBF	Radial Basis Function
RSME	Residual Mean Square Error
TCO	Traditional Collaborative Optimisation
UDC	Urban Driving Cycle

Chapter 1 Introduction

To meet increasing customer expectations of good fuel economy and driveability whilst simultaneously meeting increasingly stringent legislative limits on emissions, a consequence of new technologies added to modern engines is increased control complexity. With more engine actuators and controls to calibrate, the engine mapping and calibration task has become significantly more involved and the task of identifying optimal actuator settings is now much more difficult. This situation is challenging for current internal combustion engines, and is only likely to get worse with the introduction of 'hybrid' technologies. More efficient optimisation strategies are required to handle the increased dimensionality of the optimisation problem and to cope with the imposed complex couplings between system control variables.

1.1 Engine Mapping and Calibration

The target of Engine Mapping and Calibration (EMC) is to find the controllable actuator settings that make an engine work with improved performance (generally higher torque with lower emissions and fuel consumption). The original EMC process dealt with a single control variable and was carried out across a grid of engine load/speed points covering the whole engine operating range based on experiments (Waters, 1972 , Blumberg, 1976). As more engine control variables were introduced so an EMC process was developed based upon the experiments by using statistical techniques to analyse engine outputs trends towards achieving the best engine performance (Rishavy, 1977). The purpose of statistical methods in EMC processes is to develop accurate engine models, which are used to find optimal actuator settings.

Model Based Calibration (MBC) (Onder and Geering, 1993 , Onder and Geering, 1995) was developed based on a statistical analysis to predict the engine performance. This process became increasingly popular and powerful engine calibration approach for modern internal combustion engines, especially since the commercial toolboxes (such as MBC toolbox in MATLAB, AVL CAMEO toolbox and so on) became available. A typical MBC process is underpinned by statistical tools such as Design of Experiments (DoE) or response surface modelling and optimisation (Pilley et al., 1994 , Roudenko et al., 2002 , Sampson, 2004). DoE techniques can be used to give a set of efficient test combinations that carry all essential information to build up the engine response models, based upon the well defined design space (such as input variables, inputs range).

Statistical engine modelling is the process of modelling engine response data as a function of engine state and actuation variables. The engine response models are then used for engine calibration optimisation. Engine calibration optimisation is the process of finding the optimal engine actuator settings that produce optimal engine performance based on engine response models covering the whole engine operating range.

1.2 Project Background

This project was sponsored by Jaguar Land Rover with the aim of addressing a number of issues relating to Diesel engine steady state calibration processes. The common approach to Diesel engine calibration is based on using local optimisation at different engine states (load/speed) and generating an engine control map with locally-optimal solutions by interpolation and extrapolation. This process does not only lead to expensive computational cost, but also results in less-than-smooth engine control maps. The engine control maps are validated on the transient testing bed and vehicle and also make the engine map smooth by manually adjust the engine variable control. Since the modification of engine control map has been made based on the local optimal solution, it leads to a failure of achieving emission targets. Because of the failure of validation and modification on transient testing bed and vehicle for the calibration of optimal solution, the engine calibration needs to be re-optimised by revising the engine actuator valid range and the engine actuator map gradient limits. This led to that the driveability calibration and robustness to environments process was delayed. The reason was that calibration optimisation process cannot produce a robust optimal solution with a single optimisation run, in terms of achieving emission productions and driveability requirements.

This situation is only getting worse as more controls are added into engine systems to meet more restrictive emissions legislations and increased driveability expectations. Based upon these issues the sponsoring company set up a number of targets as follows:

- Develop a future-proof computational methodology for engine calibration optimisation, which has many inputs (design variables), typically more than 100.
- The (mathematical) methodology must be capable of expansion as engine complexity increases.

1.3 Aim and Objectives

The aim of this project is to study and implement the advanced mathematical optimisation methodology to the Diesel engine calibration optimisation problem with steady state engine test data. The selected optimisation methodology should provide computational efficiency with a clear optimisation structure and capability of expansion as engine (or powertrain) complexity increases.

In order to meet this research aim, research objectives have been determined as follows:

- Carry out an outline review from literature of existing engine calibration approaches;
- Investigate existing optimisation methodologies from literature;
- Study the present calibration approach in order to understand the issues and challenges;
- From the study of optimisation methodologies review, attempt to formulate the steady state Diesel engine calibration problem based on the reviewed optimisation methodologies;
- Test the optimisation formulation and analyse the optimisation application for Diesel engine calibration;
- Effort on the further development based on the implementation

1.4 Methodology

Figure 1.1 indicates the process flow of this research project, divided into two parts which are research preparation and research development.

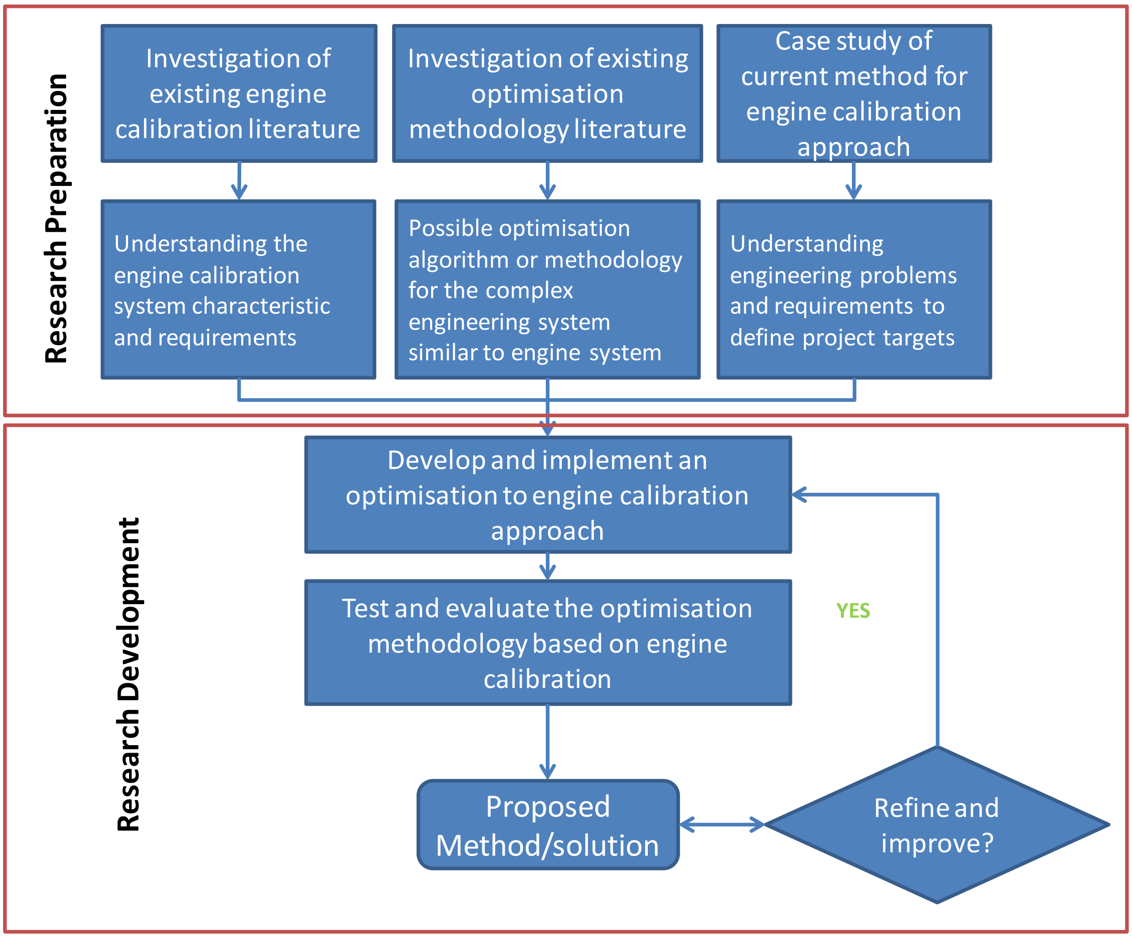


Figure 1.1 Flowchart of research methodology

Based on the research objectives, literature reviews of both engine calibration processes and mathematical optimisation methodologies are required in order to (i) understand the state of art in both engine calibration and optimisation methodology and (ii) to find opportunities to apply more advanced optimisation methodologies to engine calibration. A reference case study of the current engine calibration approach is necessary in terms of identifying the development targets of this research. Based on reviews of literature, optimisation methodologies are taken forward towards developing the engine calibration approach. The proposed calibration approach is then validated and compared with the present engine calibration approach. From the evaluation, decisions on taking further of development are taken.

1.5 Outline of Thesis

This thesis is structured as follows:

Chapter 2: literature review of optimisation. This chapter carries out a review of optimisation methodologies available for solving complicated engineering systems optimisation problems.

Chapter 3: Engine mapping and calibration literature review. In this chapter a critical review of engine mapping and calibration approaches is undertaken. Different calibration approaches are compared and discussed in order to define the research direction.

Chapter 4: Case Study of Current Engine Mapping Calibration Process. In this chapter the present Diesel engine calibration approach is studied to understand the difficulties and define the problem. A discussion and analysis of the calibration approach associated with potential optimisation methodologies is provided.

Chapter 5: All At Once (AAO) describes the first approach developed for the present Diesel engine calibration optimisation approach. The engine calibration optimisation problem is formulated using the AAO framework and the optimisation tests are carried out. A discussion and analysis of the results is provided, in contrast with the benchmark results obtained from using the conventional “current” process.

Chapter 6: Collaborative Optimisation demonstrates an engine calibration optimisation process that is formulated into a framework. The tested results are compared with AAO and present-process results. Based on the comparison and discussion, an analysis of the Collaborative Optimisation formulation is carried out to define the further development.

Chapter 7: Hybrid Collaborative Optimisation (HCO) is the further improvement of this research. The engine calibration optimisation problem is reformulated with the HCO that is specially designed for a Diesel engine calibration optimisation problem. The HCO approach results have been presented and discussed in Chapter 7.

Chapter 8: Discussion and Conclusions comparing the different calibration optimisation approaches. The decision of choosing the optimal actuator setting is made.

Chapter 9: Summary and Future work. This chapter summarises this research and provides suggestions and recommendations for future research in this area.

Chapter 2 Literature Review: Optimisation Methods and Frameworks

2.1 Introduction

This chapter presents a review of optimisation problems, optimisation algorithms (gradient-based methods and population-based algorithms), and Multidisciplinary Design Optimisation (MDO) frameworks for complex system design. A number of algorithms and frameworks are compared and analysed to assist with the selection of possible options for Diesel engine calibration optimisation.

2.2 Optimisation Problem

In the literature (Deb, 2004 , Rao, 2009 , Ravindran et al., 2006 , Reklaitis et al., 1983b) an optimisation problem is generally formulated as:

Minimise: $\mathrm{F}(\mathrm{x_i})$ $\quad i = 1, \ldots, \mathrm{I}$

Subject to: $\mathrm{LB_i} < \mathrm{x_i} < UB_i$

$\mathrm{G_j}(\mathrm{x_i}) \leq 0$ $\quad \mathrm{j} = 1, \ldots, \mathrm{J}$

$\mathrm{H_k}(\mathrm{x_i}) = 0$ $\quad \mathrm{k} = 1, \ldots, \mathrm{K}$ ***Equation 2.1***

Engineering optimisation problems are generally expressed as the minimisation of a function $\mathrm{F}(\mathrm{x_i})$ of a set of controls $x = (\mathrm{x_1}, \mathrm{x_2}, \ldots, \mathrm{x_i})$, whose values are restricted to satisfy Jj inequalities $\mathrm{G_j}(\mathrm{x_i}) \leq 0$, and a set of K equalities $\mathrm{H_k}(\mathrm{x_i}) = 0$, within variable domain bounds $\mathrm{LB_i} < \mathrm{x_i} < UB_i$.

The growth both in computing hardware development and the use of complex statistical modelling has led to the development of optimisation theories and algorithms. Figure 2.1 summarises a basic classification of optimisation algorithms (Li and Wood, 2010):

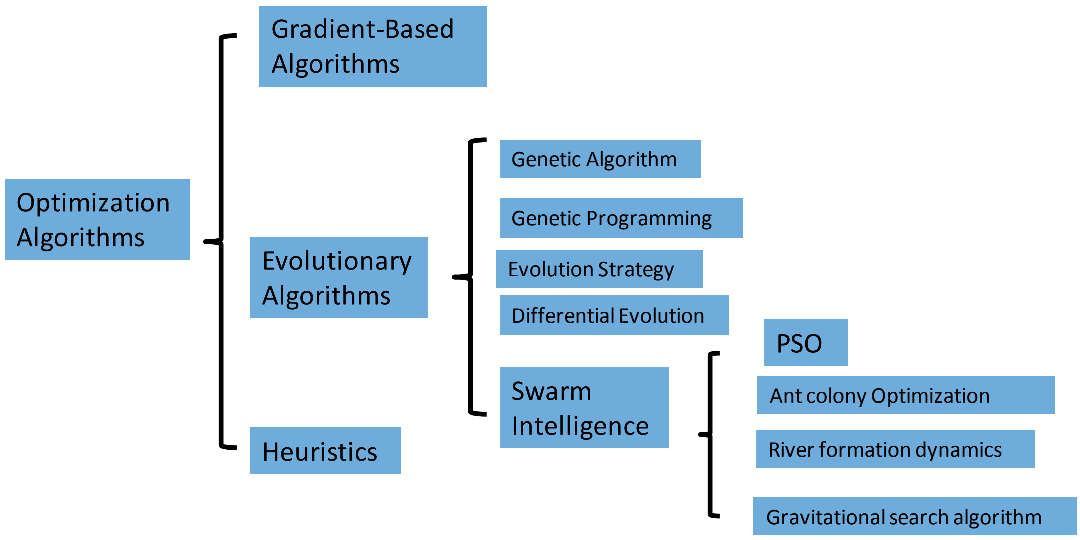


Figure 2.1 Description of search algorithm developments (Li and Wood, 2010)

2.2.1 Gradient-based Algorithms

Gradient-based algorithms use a functional gradient to find the minimum or maximum point of a function. Whilst these are rapidly-converging algorithms, they can only find a local solution (Bhatti, 2000 , Deb, 2004 , Fletcher, 1987 , Papalambros and Wilde, 2000 , Reklaitis et al., 1983b). There are a large number of gradient-based search algorithms, including Cauchy's method (Goldstein, 1962), Newton's method (Grippo et al., 1986), and so on. Cauchy's method provides rapid reduction of the objective value, particularly when the starting point is far from the optimum solution. Cauchy's method determines the gradient direction by the first order derivative and applies the steepest descent method to find the 'most local descent', based on the calculated x^{k+1}, which is shown as:

$$x^{k+1} = x^k + \propto^k * s(x^k)$$

Equation 2.2

Where: x^k : current estimate solution;

$\propto^k$ step length parameter

$s(x^k)$ search direction in the N space of the design variables x_i, *i=1,2...,N*

Cauchy's method provides a good reduction of the objective value, especially when x^0 is far from the optimum solution (Goldstein and Price, 1967). Since the higher order derivative information has been ignored, Cauchy's method is only good at finding the local optimum solution (Goldstein, 1962 , Ravindran et al., 2006). In order to construct a more global strategy, Newton's method employs higher order derivative information on the objective function based upon Taylor series expansions (Reklaitis et al., 1983a).

Due to most engineering optimisation problems involving a number of constraints, both in the design and the decision spaces, techniques to convert the constrained optimisation problem into an equivalent unconstrained optimisation problem are required. Two basic techniques are Lagrange Multipliers (LM) and Kuhn-Tucker Conditions (KTC) (Bhatti, 2000 , Deb, 2004 , Fletcher, 1987 , Papalambros and Wilde, 2000 , Reklaitis et al., 1983b). Both techniques convert the constrained problem to an unconstrained problem by combining the objective function and the constraints function into a single objective function with a number of coefficients (corresponding to the number of constraints), and then solving a set of differential equations to estimate the coefficients. The difference between LM and KTC is that KTC can handle equality and inequality constraints whereas LM can only handle equality constraints (Bhatti, 2000 , Cohon, 1985 , Ravindran et al., 2006 , Reklaitis et al., 1983b).

Sequential Quadratic Programming (SQP) (Bhatti, 2000 , Boggs and Tolle, 1995 , Ravindran et al., 2006 , Byrd et al., 1992) has been widely used to solve optimisation problems containing non-linear constraints and it represents a state-of-the-art of non-linear programming methods. SQP is based upon gradient-based methods and a KTC formulation to solve the constrained problem. As a gradient-based method, SQP also involves two major definitions from search processes, which are step size and search direction (Boggs and Tolle, 1995 , Byrd et al., 1992).

2.2.2 Population-based Algorithms

2.2.2.1 Introduction

Since the 1970s computing power has dramatically improved. Population-based computational methods were introduced in order to produce global optimal solutions within complicated design spaces (Holland, 1975 , Conley, 1980 , Conley, 1984 , Dickman and Gilman, 1989). Genetic Algorithms (GAs) are a class of search algorithms based upon the principles of evolution and natural selection. GAs do not only allow the population to indicate the search direction through the evaluation of each design in the population, but also let the solutions in the population carry the better genes with them by inducing off-spring solutions with selected parental solutions (Deb, 2001).

2.2.2.2 Principles of Genetic Algorithms

Figure 2.2 shows the flow of a standard GA process that starts with an initial population that is randomly created within the design space (Weile and Michielssen, 1997). Each design in the population set is evaluated and randomly compared between a small number of solutions by a selection function, and the selected solutions are used as 'parents' to generate new solutions as a new population (child solutions) by using crossover and mutation operators. The population of child solutions is checked against convergence criteria, and the optimisation is terminated if the convergence criteria are achieved. Otherwise, the same procedure continues until convergence criteria are satisfied.

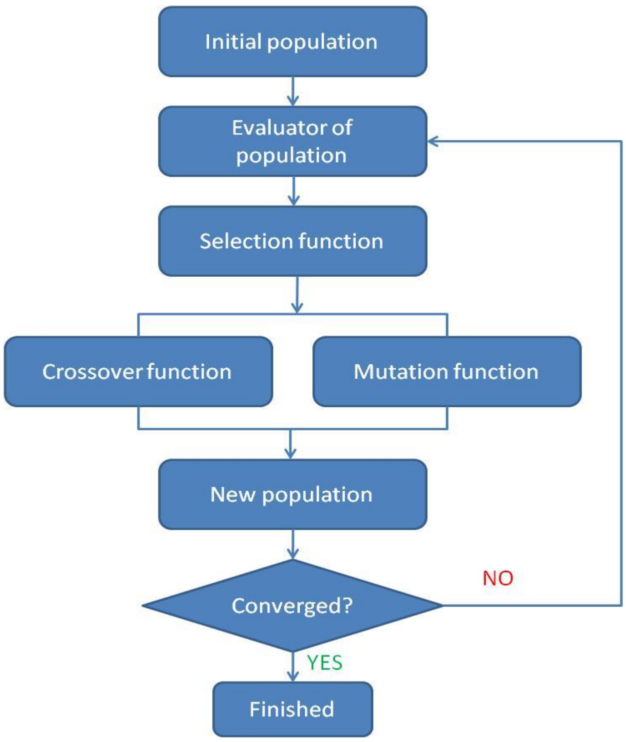


Figure 2.2 Flowchart of a standard GA process (Weile and Michielssen, 1997)

Selection operator

The selection operator is used to assign preference to the better individual solutions in a population and to allow them to pass their genes to the next generation. There are a number of ways to duplicate good solutions and eliminate bad solutions. Some of the commonly-used methods are ranking selection, roulette selection tournament selection(Goldberg, 1989b).

Ranking selection evaluates and ranks the fitness of each solution. Thereafter a number of parents are selected from the top of the rank. This selection results in a fast convergence searching route. However, the search direction concentrates in a particular area after just a few generations. Some areas are missed, which leads to a local optimal solution(Schaffer, 1985a)

Tournament selection randomly selects and compares a number (tournament size) of solutions. The tournament size must be at least two. The best individual of the tournament is chosen and placed in the mating pool as a parent solution according to its fitness value (Goldberg, 1989b).

Crossover operator

The crossover operator creates new solutions that carry genes from a parent solution from the previous generation (Bramlette, 1991). Figure 2.3 shows an example of crossover. Two solutions are selected as parent solutions by the selection operator. As survivors from selection they are assumed to be carrying better genes. According to biology and evolutionary theory, there is a chance to make a “better” child by combining the genotype of two naturally selected survivors. From an engineering point of view, there are more possibilities of finding better solutions around those two selected solutions with better fitness. User defined parts of the string are exchanged between the two parent strings. This process creates two new strings, the so called offspring or child solutions (Goldberg, 1989a).

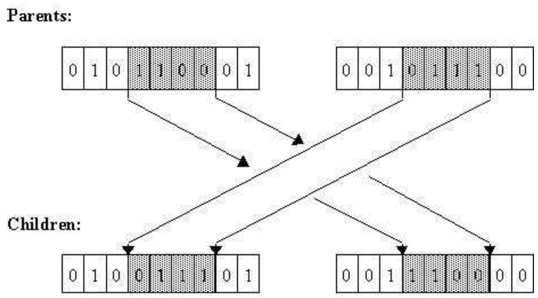


Figure 2.3 Demonstration of crossover with binary strings(Goldberg, 1989a)

In most engineering problems, real numbers are normally used for the input and output variable values. There are a number of commonly-used crossover functions in engineering applications that are intermediate and heuristic.

Mutation operator

The mutation operator is designed according to the theory of geno-variation. The essence of mutation is to discover more areas of the design space that have not been previously explored. The discovery of a new area provides the possibility of finding an ‘improved’ optimal solution. The principle of the mutation operator application with a binary number is shown in Figure 2.4. The number at the user selected bit is mutated from generation to generation (Goldberg, 1989a).

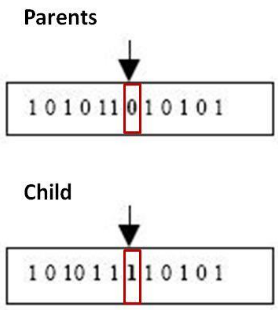


Figure 2.4 Demonstration of mutation with binary strings(Goldberg, 1989a)

Similarly, in engineering applications the mutation operator has been developed into a number of different approaches, for example Gaussian mutation (Fogel and Atmar, 1990), Uniform mutation (Houck et al., 1995), and Adaptive Feasible mutation(Srinivas and Patnaik, 1994).

Gaussian mutation adds a random number to each entry of the parent solution vector. The random number is generated according to the Gaussian distribution with a mean of zero and a standard deviation determined by user-defined parameters which are scale and shrinks. The mutated solution is evaluated as equation 2.2 (Fogel and Atmar, 1990).

$$mutated\_child_{k+1} = parent_k + (scale_k * range * (1 - shrink * \frac{k}{number\ of\ generation}))$$

Equation2.2

k is the current generation number. "Scale" controls the standard deviation of the mutation at the first generation, which is scale multiplied by the range of the initial population (which is specified by the Initial range option). "Shrink" controls the rate at which the average amount of mutation decreases. The standard deviation decreases linearly so that its final value equals 1 –“Shrink” times its initial value at the first generation (MATLAB User Guide). For example, if “Shrink” has the default value of 1, then the amount of mutation decreases to 0 at the final step.

Uniform mutation returns the mutated child solution by replacing each of the individual solutions of selected entry (defined as a number of variables or a part of solution) with the probability rate to be mutated. A random number is used to replace the selected byte, which is uniformly selected from the corresponding range of each entry. Similarly, adaptive feasible mutation uses the same procedure with respect to both linear and non-linear constraints (Houck et al., 1995).

2.2.2.3 Multi-Objective Optimisation Problem (MOOP):

As engineering systems are becoming more complicated, so too are the optimisation processes required to be able to handle such complexity. In these engineering optimisation problems the traditional single objective optimisation cannot satisfy the engineering system requirements anymore, therefore such a type of engineering system is formulated as a Multi-Objective Optimisation Problem (MOOP) (Ravindran et al., 2006). A typical MOOP is presented in Equation 2.3 (Schaffer, 1984; Fonseca and Fleming, 1995):

Minimise: $F_i(\vec{x})$ $\quad\quad$ $\mathrm{i} \in (1,2, \dots \mathrm{m})$

Subject to: $g_j(\vec{x}) \leq 0$ $\quad\quad$ $\mathrm{j} \in (1,2, \dots \mathrm{n})$

$h_k(\vec{x}) = 0$ $\quad\quad$ $\mathrm{k} \in (1,2, \dots \mathrm{p})$

Equation2.3

m is the number of objective functions F_i, and $\vec{x} = [x_1, x_2, ... x_r]^T$ is the vector of design variables. The optimal solution is defined as a set of $\vec{x}$ that yields the optimum values of all the objective functions and that satisfies the inequality constraints $g_j(\vec{x}) \leq 0$ and equality constraints $h_k(\vec{x}) = 0$.

To compare, for a single optimisation problem using one solution to represent the optimal solution, it will be rare that a single point will simultaneously optimise all the objective functions (Coello, 2006). Therefore, 'trade-offs' are normally expected while dealing with multi-objective optimisation problem. The set of solutions of a multi-objective optimisation problem consists of the decision vectors $\bar{x}_s$ that cannot be further improved in any objective dimension without retrogression in another objective (Zitzler and Thiele, 1998). These decision vectors in the design space are termed the Pareto-optimal, non-dominated solution, trade-off or non-inferior solution (Horn, 1997).

In theory any multi-objective optimisation problem can be converted into a single-objective optimisation problem. There are two common ways to convert a multi-objective optimisation problem to a single-objective optimisation:

i. Aggregate all the objectives into one scalar function and treat all the objectives apart from the most important one as constraints with given limits (Coello, 2002; Fonsesca and Fleming, 1994);
ii. The weighted sum method was the most common procedure used in early approaches to solve the multi-objective optimisation problem. It assigns a weight w_i to each objective f_i and the combination of all these weighted objectives is formulated as (Coello, 2002):

$$F(x) = \sum_{i=1}^{n} w_i \times f_i$$

The weighted sum approach produces a single compromise solution without requiring any further information from the decision maker after the weights are established. This procedure only produces one single-point optimal solution for each single run. If the candidate ‘optimal solution’ is not acceptable or the weighting information is updated, additional simulation runs would be required until a suitable solution was found (Fonsesca and Fleming, 1994). Although a suitable solution may be found after many simulation runs, this single optimal solution is not necessarily the best choice. Furthermore, the weighted sum method cannot find the real Pareto-Frontier optimal solution set in a non-convex objective space (Deb, 2001)

Several extensions of similar scalarised sum weighted approaches have been reported in the literature (Koski and Silvennoinen, 1987; Steuer, 1989; Das and Dennis, 1997).Jin et al. (1993) discussed how to generate a Pareto-optimal solution set by using the weighted sum method with dynamic weights, also called dynamic weighted aggregation. These methods emphasise the selection of non-arbitrary and efficient weights and can produce an approximate Pareto-optimal solution. As the weight becomes more efficient, knowledge and preference information of the system is required from the decision maker. These methods are more computationally expensive (Marler and Arora, 2004).

2.2.2.4 Evolutionary Algorithms

Table 2.1 summarises a number of major developments in evolutionary algorithms, which are reviewed in this section.

Table 2.1 Summary of evolutionary algorithms

Algorithms	Technique	Advantage	Disadvantage	Reference
VEGA	Vector evaluation	Addresses all objectives simultaneously	the optimal solution of each single objective is not well covered	(Schaffer, 1985b , Schaffer, 1985a)
MOGA	Non-dominated ranking; sharing function used in objective space	Able to produce non-dominated solution set; less crowded distribution in objective space	Depends on an appropriate selection of sharing function	(Fonseca and Fleming, 1993 , Fonseca and Fleming, 1998 , Sinoquet, 2009)
NSGA	Non-dominated ranking; sharing function used in design space	Non-dominated solution set can be produced; maintains diversity of Pareto-solution	Depends on parameter of crowding distance	(Goldberg, 1989b , Srinivas and Deb, 1994)
NPGA	Tournament comparison ranking scheme; niching technique performed over population	Convergence rate is controllable by user defined parameter, diversity maintained over whole population	Elitism solutions are not maintained from generation to generation; depends on the parameter t_{dom};	(Horn et al., 1994)
Rudolph's MOEA	Elitist algorithm based on non-dominated ranking; number of optimum solutions saved externally	Fast convergence rate; diversity maintained over whole population	Depends on parameter too much; computation is expensive	(Rudolph, 1999)
NSGAII	Elitist algorithm based on NSGA; sharing function performs in objective space	Fast convergence rate; diversity controlled by crowding distance	Depends on parameter of crowding distance	(Deb, 2001 , Deb, 2004)
SPGA	Elitist algorithm performed by storing Pareto optimal solution externally; uses clustering	Reduces number of iterations; maintains diversity of Pareto-solution	Storing Pareto optimal solution and clustering algorithm costs	(Zitzler et al., 2001 , Zitzler and Thiele, 1998)

Vector Evaluated Genetic Algorithm

Schaffer (1984) was the first to find a Pareto-optimal solution set for a multi-objective optimisation problem by using an Evolutionary Algorithm (EA). Schaffer's (1985) multimodal EA approach, also called the Vector Evaluated Genetic Algorithm (VEGA), is possibly the simplest Multi-Objective Genetic Algorithm (MOGA) extension from a single-objective genetic algorithm. Schaffer (1985) proposed that a crossover between two higher ranking solutions corresponding to different objectives could find good trade-off solutions between the two objectives. Since each sub-population had been ranked with a different objective, and the solutions of higher rank were selected to generate a mating pool, this procedure meant that the solutions of each sub-population converge to the optimal solution of just one particular objective. Thus, the VEGA approach results in a set of solutions that can be distributed around the extreme value of different objective optimal solutions in the objective space (Coello, 1999). The shortcoming of this approach is that many solutions are missed on the approximate Pareto-frontier between the different individual objective optimal solutions (Deb, 2001).

Since VEGA was introduced, many people have tried to update it to obtain a better distribution on the Pareto-frontier. Schaffer (1985) suggested that the diversity of solutions could be maintained by emphasising non-dominated solutions in a population.

Non-dominated Sorting Genetic Algorithm (NSGA)

Goldberg's (1989) non-dominated sorting concept in GA was implemented again by Srinivas and Deb (1994). To maintain diversity in the design space, a sharing function with the niche technique has been used to assign the shared fitness to every individual in the same non-dominated set. Tournament selection based upon the shared fitness has been used to produce a mating pool. Crossover and mutation operators are used as normal. In the NSGA process, the diversity of a non-dominated solution set is improved by assigning a rank to each individual within the non-dominated solution set. Also, the share function has been carried out again and the “niching” has been performed in the design space. Therefore this process provides a better possibility of finding more local optimal solutions.

Multi-Objective Genetic Algorithm (MOGA)

Many attempts have been made to improve the VEGA algorithm. Fonseca and Fleming (1993) represented a Multi-Objective GA (MOGA) that, for the first time, used the non-dominated classification of a GA population to solve a multi-objective optimisation problem. The difference between MOGA and the standard GA revolves around the way in which fitness is assigned to each solution in the population. The idea of MOGA is that all the individuals can be sorted into different non-dominated sets by assigning a rank to them. In order to maintain the diversity of non-dominated solution set, Fonseca and Fleming implemented Goldberg’s (Goldberg et al., 1991 , Miller and Shaw, 1996) idea of a sharing function for a “niching” mechanism by selecting individuals from the same rank solution set (Fonseca and Fleming, 1998).

Niched-Pareto Genetic Algorithm (NPGA)

Horn and Goldberg (Horn et al., 1994) employed a method to solve the multi-objective optimisation problem, which was a Pareto dominated tournament instead of a non-dominated ranking method. They proved that the amount of selection pressure and convergence speed can be controlled by adjusting the size of the tournament. To maintain the diversity between the individuals, Goldberg (1989) suggested that the individuals are degraded by simply dividing the objective fitness by the niche count to find the shared fitness. Hence, the individuals within the share distance σ_{share} of each other degrade each other's fitness. Since tournament selection has been used instead of non-dominated ranking, the optimiser cannot be certain that all the selected solutions are in the non-dominated solution set. The quality of the selected solutions depends heavily upon the randomly selected solution size t_{dom} (Deb et al., 2000).

Rudolph's Multi-Objective Evolutionary Algorithm (Rudolph, 1999)

Early evolutionary algorithms had the disadvantage that they could lose better solutions in the population from previous generations. A 'remedy' was proposed by Rudolph (1999), Deb (2000) and Zitzler (Zitzler and Thiele, 1998). Rudolph (1999) proved that GAs converged to the global optimal solution better by keeping the elitist solution in the population from generation to generation. Rudolph (1999) suggested a complete procedure for a GA based Multi-Objective Evolutionary Algorithm with elitism. Rudolph's suggestion represents an elitist multi-objective optimisation algorithm: every elitist solution in the population is preserved in the next generation. Nevertheless, Rudolph's method does not involve any diversity preservation mechanisms, and it produces the final Pareto-optimal solution set distributed only around the different objective optimal solution (similar to VEGA).

Non-dominated Sorting GA II (NSGA II)

Following Rudolph (Rudolph, 1996) the elitist algorithm has been examined by many researchers. The most innovative algorithm is the elitist non-dominated sorting genetic algorithm (NSGAII). Deb (2000) suggested preserving the elitism solution from generation to generation and maintaining the diversity between the solutions at the same time by applying NSGA. In order to keep the elitism in the population from generation to generation, for every generation the parent solutions are combined with offspring solutions. Then the selection procedure is carried out with this combination of solution sets according to the rank of the non-dominated solution set. To choose the solution from the same non-dominated solution set, a crowding distance is calculated to assign selecting rank for each solution. The elitist mechanism has been implemented in NSGAII by combining the parent and offspring population to preserve the elitism from generation to generation. The diversity between the non-dominated solutions is retained by using tournament selection of crowding distance comparison (Deb et al., 2000). On the other hand, the crowding distance assignment does not require any extra parameters, for example σ_{share} for sharing function in MOGA, NSGA and NPGA.

Strength Pareto Evolutionary Algorithm (De Jong and Spears):

A number of elitist multi-objective EAs have been recently proposed, such as Zitzler and Thiele's (Zitzler and Thiele, 1998) Strength Pareto Evolutionary Algorithm (De Jong and Spears). SPEA stores the Pareto-optimal solutions externally so that the Pareto-optimal solutions are not lost from generation to generation. The clustering algorithm has been used in SPEA to produce a good distribution of Pareto-optimal solutions on the Pareto-front. In order to prune the external non-dominated elite solution, while keeping the characteristics of the original set, a clustering algorithm has been implemented in the SPEA optimisation process (Zitzler and Thiele, 1998). A non-dominated set of p elements is partitioned into q groups of relatively homogeneous elements by cluster analysis, where $p > q$. One solution is chosen from each of these groups to be used as parent solution. In SPEA, any solution in the Pareto-optimal front will immediately get stored in the external population, and will not be deleted until a solution is found which can dominate it or perform better than it in the same cluster (Zitzler et al., 2001). Clearly, the external population stores the elitism from generation to generation to preserve them. Also, a good spread from the non-dominated solution set is generated by the clustering algorithm and the concept of non-domination. However, the clustering algorithm uses less parameter than the sharing function. It still introduces a parameter $\overline{\mathrm{N}}$, which is the size of the external population. The success of SPEA depends upon how this parameter is defined (Zitzler et al., 2001).

2.3 Multidisciplinary Design Optimisation (MDO)

Another approach to considering the complex process of engineering system design involves working with a number of various interacting disciplines. Based on these natural characteristics, MDO methodologies have been developed to deal with these complex engineering systems. MDO aims to explore and perform optimisation at every discipline while following “interaction” rules. MDO frameworks have been used to optimise the design of aircraft (Kroo et al., 1994), automotive vehicles (Kim et al., 2001), and aerospace vehicles (Braun and Moore, 1996).

The original MDO methodology was developed in the aircraft industry to improve the design of complex aircraft systems that involve the work of many specialists in various dependent disciplines (Kroo et al., 1994).

In complex system MDOs, there is always the challenge of balancing computation and organisation. A good MDO strategy is required to (Kroo et al., 1994):

- decrease the dimensionality;
- simplify the analysis expense;

- maintain the consistency of the whole system with each discipline;
- properly explore the design space of each discipline for a good search.

Multidisciplinary optimisation methodologies for complex engineering systems design are developed based upon the different formulation of the optimisation problem. Different formulations of an optimisation problem lead to the development of different MDO frameworks. The methodologies of MDO can be divided into single-level system (e.g. All At Once and Multidisciplinary Feasible Method) and multi-level system (e.g. Collaborative Optimisation, Bi-Level Integrated System Synthesis and Analytical Target Cascading), which are reviewed in this chapter. Table 2.2 summarises the MDO frameworks that are reviewed.

Table 2. 2 Summary of Reviewed MDO Frameworks

	MDO	Advantages	Disadvantages	Reference
Single Level Strategy	AAO	Simple to use and capable for all MDO problem (both hierarchical and non-hierarchical)	Expensive computation costs, the performance depends on the number of disciplines and variables very much	(Balling et al., 1994 , Cramer et al., 1992 , Kodiyalam and Center, 1998 , Cramer et al., 1994).
	MDF	Coordinate all the disciplines with a single level optimiser, no restrictions on data communication between disciplines (non-hierarchical coupling)	Extremely expensive computation is required for each iteration as more disciplines involved, low robustness to find optimal solution	(Braun et al., 1996 , Cramer et al., 1994)
	IDF	Less computation required and better robustness compare with MDF,	Does not ensure the consistency, the dimensionality increased by decomposing coupling variables	(Allison et al., 2005 , Cramer et al., 1994 , Allison, 2004)
Multi-Level Strategy	CO	The decomposition provides the maximum possibility to find the local optimal solution and reduce the data communication requirements	With the increased dimensionality, it is difficult to decompose the system into disciplines	(Kroo et al., 1994 , Kroo and Manning, 2000 , Braun and Kroo, 1995 , Braun and Moore, 1996 , Sobieszczanski-Sobieski and Haftka, 1997)
	BLISS	Better performance when design variable cluster into few relative design variable groups and disciplines have large number of local variables.	Difficulty with nonhierarchical strongly coupled MDO problem with less flexibility in the local design space	(Sobieszczanski-Sobieski et al., 1998 , Sobieszczanski-Sobieski and Haftka, 1997)
	ATC	Clear and efficient structure for hierarchical MDO problem	Difficulty to handle nonhierarchical MDO problem	(Kim et al., 2001 , Michelena et al., 1999 , Papalambros, 2001 , Papalambros, 2002 , Allison et al., 2009 , Kim, 2001)

2.3.1 Single-level MDO Strategies

Single-level MDO strategies aim to solve a problem with a number of disciplines in which all the requirements of expertise can be easily acquired by full control of inputs from the optimiser. Therefore, the analyser may create a new discipline by focusing upon the interaction between disciplines. Also, the shared variables between disciplines have to be obtained simultaneously. Figure 2.5 shows an example of a system design with a number of fully-coupled disciplines (or subsystems). The single-level MDO strategies aim to solve the total optimisation problem by handling all the disciplines together (Allison et al., 2005).

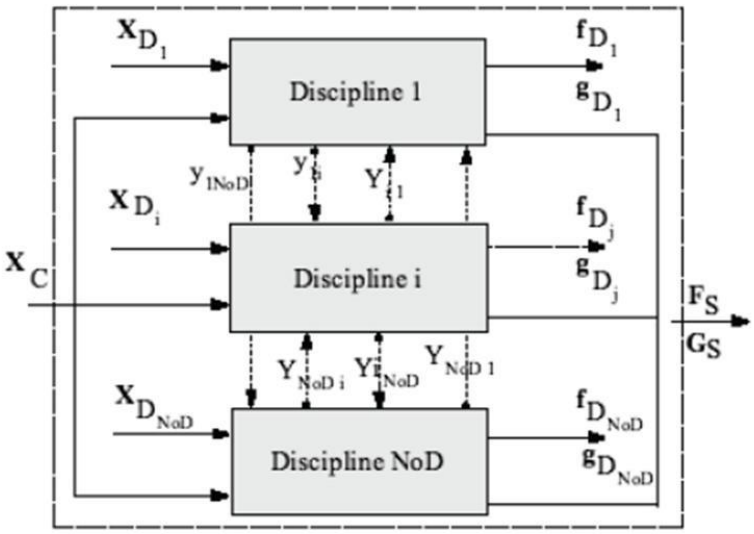


Figure 2.5 Demonstration of MDO Problem with Fully Coupled Disciplines (Allison et al., 2005)

All At Once (AAO)

The All-At–Once strategy (AAO) is a highly centralised approach. In the AAO approach, the system optimiser deals with three sets of design variables: the original design variables, the coupling variables, and the state variables (Balling et al., 1994 , Cramer et al., 1992 , Kodiyalam and Center, 1998). The values of these design variables are evaluated by the optimiser of the AAO in the design space. The values of the original design variables are reproduced in the optimal objective function values; the values of coupling variables satisfy the auxiliary constraints to ensure the consistency of the system; the values of state variables satisfy the governing equations. Figure 2.6 demonstrates the AAO framework where decision space design, system analysis, and subsystem analysis are all performed simultaneously (Allison, 2004).

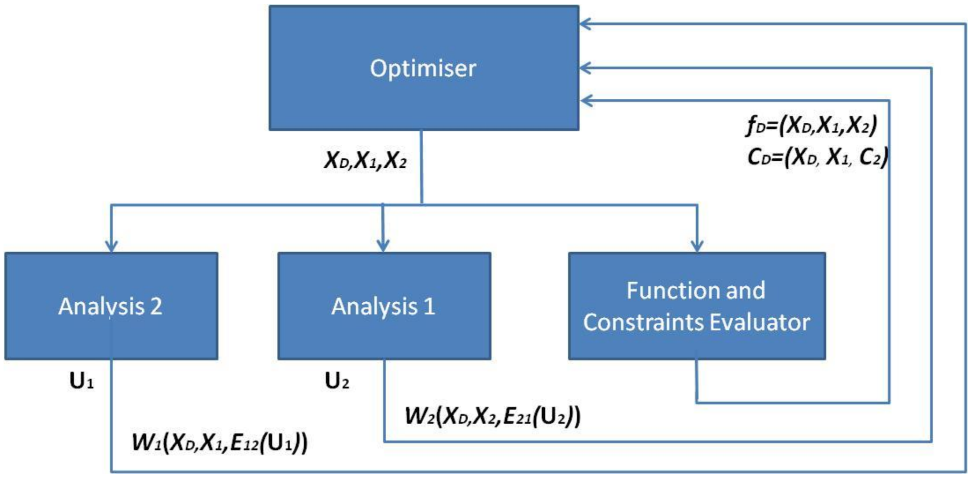


Figure 2.6 Illustration of AAO Framework

AAO is the most basic of MDO methodologies and has been commonly used in industry, although it is restricted to small design problems. Because AAO takes all the local optimisations away and gives full control of all the design variables to the system level optimiser, this leads to an expensive computation cost. It ensures that MDO is performed and a global objective is met by performing the optimisation at all the disciplines and controlling the entire engineering system design with one optimiser. Such an optimisation can immediately determine how changes to one particular part of the system affect the whole system.

Multidisciplinary Feasible (MDF)

Multidisciplinary feasible (MDF) approaches use a single system-level optimiser to coordinate all of the subspace analysers (Cramer et al., 1994). In MDF approaches, the system analyser is responsible for finding the optimal solution in the design space using the optimiser with the appropriate response functions. The system analyser is nested in order to find the optimal solution at every optimisation iteration. Kodiyalam and Center (Kodiyalam and Center, 1998) summarised a MDF formulation to a general optimisation problem (Equation 2.6):

Minimise: $F(X_D, U(X_D))$

Subject to: $g\big(X_D, U(X_D)\big) \leq 0$ and boundaries on design variables X_D

Equation 2.6

In this formulation X_D is the vector of design variables and $U(X_D)$ is the output of the Multidisciplinary Analysis (MDA). X_D and $U(X_D)$ are used to evaluate the objective $F(X_D, U(X_D))$ and constraints $g\big(X_D, U(X_D)\big)$. Figure 2.7 shows the structure and data flow of MDF. In Figure 2.7 $U_1(X_D)$ is used to evaluate the output of discipline1, where X_D is the input of discipline1 that is evaluated by $F_{12}(U_2(X_D))$; E_{12} is the transfer function of discipline 1 that the form is suitable for use by discipline 1.

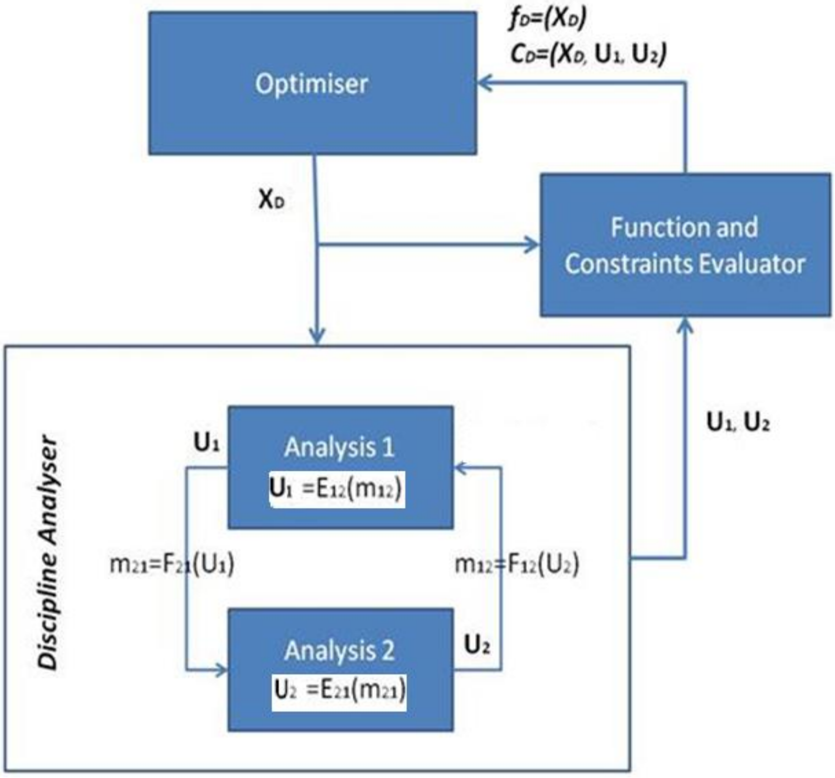


Figure 2.7 Demonstration of Data Flow under MDF Structure (Allison et al., 2005)

The MDF approach is only concerned with treating non-hierarchical problems. During the MDF process there are no restrictions on data communication between the subspaces. However, sometimes the large iteration loops of MDF can be computationally expensive, especially when more subsystems are involved and the system analysis requires more iteration to converge. Due to the fact that subsystem analysis cannot be carried out in parallel in the MDF approach, MDF exhibits low robustness for finding the optimal solution when the system analysis does not converge to one set of design settings (Braun and Moore, 1996).

Individual Disciplinary Feasible (IDF)

Compared to MDF, the Individual Disciplinary Feasible (IDF) approach was developed in order to improve the limitations of the MDF approach. The difference between MDF and IDF is that the optimiser of IDF coordinates the interactions between the subsystems, rather than relying on the simple iterations of MDF analysis (Allison, 2004). IDF allows the optimiser to directly drive the controls of individual disciplines in order to achieve multidisciplinary feasibility and optimality, whilst maintaining individual discipline feasibility. From literature (Allison et al., 2005 , Balling et al., 1994 , Braun et al., 1996 , Cramer et al., 1992 , Cramer et al., 1994), IDF can be summarised by Equation 2.7:

Minimise: $F(X_D, U(X))$ with respect to $X = (X_D, X_\mu)$

Subject to: $g\big(X_D, U(X)\big) \leq 0$

$C(X) = X_\mu - \overline{m} = 0$ and boundaries on design variables$X$

Equation 2.7

$F(X_D, U(X))$ is the total objective function and X is the vector of design variables that consists of X_D and X_μ. X_D is the vector of state variables and X_μ is the vector of interdisciplinary coupling variables. $g\big(X_D, U(X)\big)$ is the constraint function and $C(X)$ describes the consistency constraints that ensure that the shared variables are consistent.

Figure 2.8 shows that the IDF approach uses a parallel process to optimise the complex system rather than simply using the same analysis tool repeatedly. In the IDF approach, the non-hierarchical links in the system are broken down by using the constraints so that the optimisation problem can be formulated in a hierarchical system (Allison, 2004).

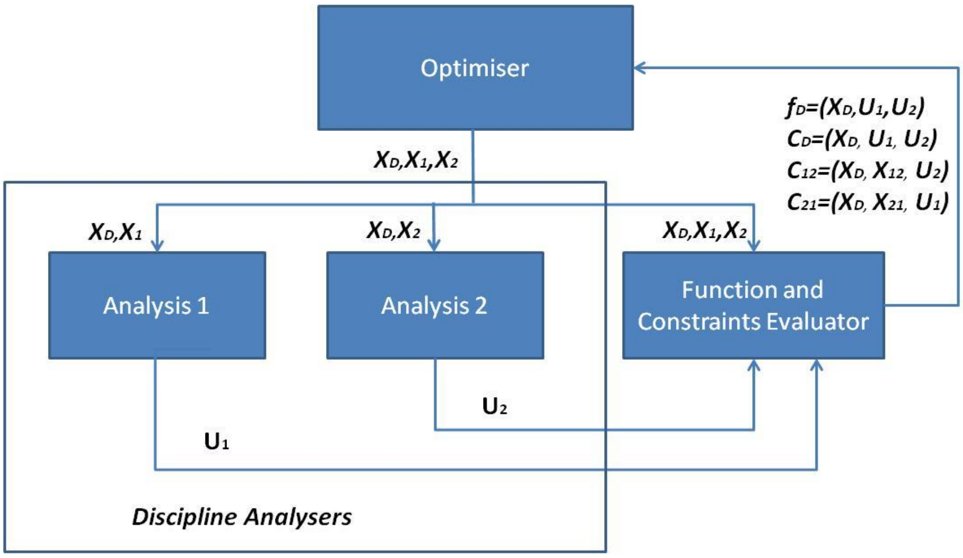


Figure 2.8 Illustration of IDF Data Flow Structure

The IDF approach does not ensure that multiple disciplines are consistent at every iteration, but only ensures that the constraints are satisfied at each discipline individually. In this process, the coupling variables become design variables in each discipline. Therefore, the dimension of the system is increased by breaking down the non-hierarchical links.

2.3.2 Multi-level strategies

With single level MDO strategies, difficulties in communication and organisation between disciplines are minimised. As design problems become more complex, the number of disciplines increases and it becomes more difficult for a centralised process (Braun and Moore, 1996). These difficulties with multidisciplinary design are particularly evident in the design of aerospace vehicles (Sobieszczanski-Sobieski and Haftka, 1997) and aircraft conceptual design (Kroo et al., 1994) Since the analysis and design tasks become more decentralised, communication requirements become more severe and more analysis needs to be handled during the process of a complex system design. A need exists, not simply to increase the ability of analyses, but rather to have a more organised structure of design optimisation problem that better explores the design problem, improves performance, and reduces complexity (Allison, 2004 , Kroo, 2004 , Sobieszczanski-Sobieski and Haftka, 1997).

The interdisciplinary coupling inherent in MDO tends to present additional challenges beyond those encountered in a single-discipline optimisation, and so Sobieski concluded that the computation, complexities and even the difficulties of organisational challenges for implementing the necessary coupling in system are increased (Sobieszczanski-Sobieski and Haftka, 1997). Faced with these challenges, single-level MDO systems are becoming incapable or inefficient. Therefore, a number of Multi-level MDO frameworks have been introduced.

Collaborative Optimisation

Braun and Kroo (1995) introduced Collaborative Optimisation (TCO, Traditional Collaborative optimisation) as a distributed system design (Braun and Kroo, 1995 , Braun and Moore, 1996 , Kroo and Manning, 2000 , Kroo, 2004). TCO is a design architecture that is capable of solving a complicated optimisation problem in any multidisciplinary analysis environment, but is specifically intended for large-scale distributed analysis applications (Braun and Kroo, 1996). In the TCO approach, an optimisation problem of the complex system is decomposed along disciplinary boundaries into a number of sub-problems, and the sub-problems are brought into multidisciplinary agreement by system-level coordination procedures. The TCO formulation is a two-level hierarchical scheme for MDO, which typically involves a system optimiser (top level) and sub-system or discipline level (bottom level), as shown in Figure 2.9.

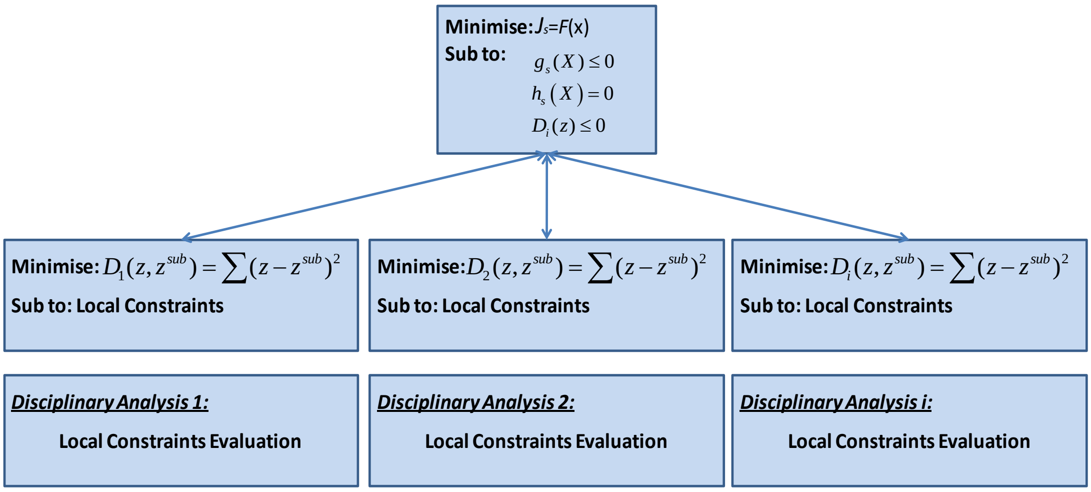


Figure 2.9 Basic Structure of Collaborative Optimisation Architecture (Kroo and Manning, 2000)

The system optimiser aims to minimise the system objective ($f_s(X)$) while satisfying the interdisciplinary compatibility D_i by varying the multidisciplinary variables X. Collaborative Optimisation (CO) at system level can be formulated as Equation 2.8:

Minimise: $J_s = F_s(X)$

Subject to: $g_s(X) \le 0$

$h_s(X) = 0$

$D_i(z) \le 0$ ***Equation 2.8***

The nomenclature is defined as follows:

$F_s(X)$: System level objective function

$g_s(X)$ and $h_s(X)$: system level constraints function

X: Vector of system level design variables

Z: Vector of interdisciplinary coupling variables, $Z \in X$

D_i: Vector of consistency term of length N

N: Number of disciplines

For every iteration the system optimiser varies the X as variables to improve the system level objective and passes Z down to the sub-system.

The subspace can then be formulated as Equation 2.9 (Braun, 1996):

Minimize: $D_i(z, z^{sub}) = \sum(z - z^{sub})^2$

Subject to: $c_i(x_i, z, z^{sub}) \leq 0$

Equation 2.9

The nomenclature is:

$D_i(z, z^{sub})$: Sub-system local objective function at discipline *i*

$c_i(x_i, z, z^{sub})$: Local constraints function at discipline *i*

x_i: Sub-system local design variables at discipline *i*

z: Vector of sub-system parameters that is passed down from system level as targets

z^{sub} : Interdisciplinary coupling variables at discipline *i*

For each iteration the improved z is sent to the subspace and is used as parameters of the sub-system. The objective in the subspace is calculated in a least squares sense as the discrepancy from the sub-system level design variables z^{sub} and system target z.

For centralised strategies such as MDF or AAO, which use one central system optimiser, data communication and structure organisation become extremely expensive computationally when the system is complicated. Collaborative Optimisation can reduce the computation of complicated system optimisation while keeping the system consistent by employing a subsystem optimiser and a unique treatment of the system level and subspace level design problem (Allison, 2004). The main aim of each subsystem optimiser is to get coupling variables consistent between subsystems, while a system optimiser provides coordination and minimises the overall objective (Braun and Kroo, 1996). Specifically, each subsystem controls a local design space variables and is determined by satisfying the local constraints. The system optimiser performs the coherence and communication between the different subsystems by managing, passing and decomposing the targets of the subsystem.

In the Collaborative Optimisation (MDO/CO) framework, the multidisciplinary optimisation problem is decomposed into disciplines at a subsystem level. In each discipline, the deviation of each variable from the target value passed down from the system level is optimised while subject to the constraints that are used to define the discipline space. In practice, the discipline optimisers minimise the distance from the local optimal solution to the global optimal solution locally in the discipline space (Braun and Kroo, 1996). In each discipline, the constraints for the other disciplines are released to provide the maximum possibility of finding the local optimal solution (Braun and Kroo, 1995). The decomposition of MDO/CO can reduce the communication requirements and the size of the overall system optimisation problem

(Kroo, 1997). However, it does increase the number of variables by creating target variables for discipline optimisers (Sobieszczanski-Sobieski et al., 1998). Although Collaborative Optimisation is a clear hierarchical MDO structure, it has been well implemented with non-hierarchical MDO problems by reorganising the data communication at the system level (Kroo, 2004).

Bi-Level Integrated System Synthesis (BLISS)

Bi-Level Integrated System Synthesis (BLISS) is a MDO method of engineering systems, which involves system optimisation with a small number of system sharing variables and a number of sub-systems having a large number of local variables compared to the system (Sobieszczanski-Sobieski et al., 1998). BLISS uses a relatively small number of design variables, which are shared with the disciplines. The solution of system level problem is obtained by using the derivatives of the state variables with respect to system level design variables and the constraint functions, which are obtained from discipline optimisations (Sobieszczanski-sobieski, 1999). The BLISS process is described as follows (Sobieszczanski-Sobieski et al., 1998):

- define initial values of global variables (Z) and local sub-system variables (X);
- conduct system analysis to compute the state variables and the design constraint function values;
- check the convergence for termination, both at system level and at sub-system level

$$Z_{new} - Z_{prev} < \delta_{global} \ \& \ X_{new} - X_{prev} < \delta_{local}$$

- conduct sensitivity analysis for both the system level and the subsystem levels by using the derivatives of the state variables/outputs of sub-system (Y_r)

and the input variables from the sub-systems r to sub-system s (Y_{rs}) with respect to global design variables (Z) and subspace design variables (X);

- at subsystem r, given X and Z, the optimisation is carried out by varying ΔX to minimise the objective function ($\emptyset_r = \frac{d\emptyset}{dX_r}^T \Delta X_r$) of sub-system r, which is expanded in a Taylor series;
- evaluate d(Φ,Z) for system optimisation, where Φ is the system objective function;
- solve the optimisation problem at the system level to get ΔZ, and update X and Z with

  $Z_{new} = Z_{prev} + \Delta Z$ & $X_{new} = X_{prev} + \Delta X$;
- repeat from step 1.

Essentially, BLISS distributes the difficulty of system design between the disciplines and uses a gradient-guided path to reach the improvement of system level design by alternating between the disciplines and the system level (Sobieszczanski-Sobieski et al., 1998). BLISS has been implemented and has been found to have performed well with the optimisation of complicated systems (Sobieszczanski-Sobieski and Haftka, 1997). Typically, the design variables cluster into a set of relatively few design variables that govern the system design; and a set of local variables that governs the local discipline design detail (Sobieszczanski-sobieski, 1999).

Analytical Target Cascading (ATC)

Analytical Target Cascading (ATC) is a methodology for product design architectures that support some type of hierarchical partitioning (Papalambros, 2001).. When the system design problem can be modelled analytically, the process can be formalized as a MDO problem referred to as ATC. The term ‘analytically’ here means any functional representation that computes the responses for some given values of the elements’ design variables, and the functional representations from lower level elements are used as design variables at upper levels (Papalambros, 2001).

In the ATC process, the system design targets are cascaded through a hierarchy of design groups (Allison et al., 2005). The solution of a partitioned system optimisation problem is obtained by solving the sub-system optimisation problem and matching their optimal solution at system level using a coordination strategy. Two types of models are used in the ATC process; optimal design model and analysis model (Lygoe, 2010 , Papalambros, 2002). The optimal design model is used to make decisions based on analysis model evaluations; analysis models take values of parameters, design variables and linking variables that are from lower level responses and return values of responses for design problems (Papalambros, 2001). A linking variable is defined as a common variable between more than one design problem. The ATC partitioning process can be carried out with the original problem by the following steps:

- minimise the discrepancy between the original problem objective and the response from the analysis model while satisfying all the constraints, shown in Equation 2.10:

Minimise: $|\boldsymbol{T} - \boldsymbol{R}(\boldsymbol{x})|$

Subject to: $\boldsymbol{g_i}(\boldsymbol{x}) \leq \boldsymbol{0}$

$$\boldsymbol{h_j}(\boldsymbol{x}) = \boldsymbol{0}$$

$$\boldsymbol{X_k^{min}} < \boldsymbol{X_k} < \boldsymbol{X_k^{max}} \qquad \textbf{\textit{Equation 2.10}}$$

The nomenclature is:

T: overall design target

R: total objective response function of x

$\boldsymbol{i\epsilon(1, 2, \ldots number\ of\ inequality\ constraints)}$

$\boldsymbol{j\epsilon(1, 2, \ldots number\ of\ equality\ constraints)}$

$\boldsymbol{k\epsilon(1, 2, \ldots number\ of\ design\ variables)}$

- cascade the system optimisation problem to the subsystem optimisation problems; at level *p* the optimisation problem of q can be formulated as:

Minimise: $\left|\boldsymbol{R_{pq}}(\widetilde{\boldsymbol{X}}) - \boldsymbol{R_{pq}^{U}}\right| + \left|\boldsymbol{y_{pq}}(\widetilde{\boldsymbol{X}}) - \boldsymbol{y_{pq}^{U}}\right| + \boldsymbol{\varepsilon_R} + \boldsymbol{\varepsilon_y}$

Subject to: $\sum\left|\boldsymbol{R_{pq}}(\widetilde{\boldsymbol{X}}) - \boldsymbol{R_{pq}^{L}}\right| \leq \boldsymbol{\varepsilon_R}$

$$\sum\left|\boldsymbol{y_{pq}}(\widetilde{\boldsymbol{X}}) - \boldsymbol{y_{pq}^{L}}\right| \leq \boldsymbol{\varepsilon_y}$$

$$\boldsymbol{g_{pq}}(\boldsymbol{R_{pq}}, \boldsymbol{X_{pq}}, \boldsymbol{y_{pq}}) \leq \boldsymbol{0}$$

$$\boldsymbol{h_{pq}}(\boldsymbol{R_{pq}}, \boldsymbol{X_{pq}}, \boldsymbol{y_{pq}}) = \boldsymbol{0}$$

Equation 2.11

$\mathrm{R_{pq}^{U}}$: target value of response from upper level

$\mathrm{R_{pq}}$: responses computed by analysis models

$\mathrm{R_{pq}^{L}}$: target value of response from lower level

$\mathrm{y_{pq}^{U}}$: target value of linking variable from upper level

$\mathrm{y_{pq}}$: linking design variables

$\mathrm{y_{pq}^{L}}$: target value of linking variable from lower level

In order to solve the multilevel engineering system design, ATC has a strong resemblance to TCO (Braun et al., 1996 , Tappeta and Renaud, 1997). However, the ATC multidisciplinary design optimisation framework is based on hierarchical organisation and analysis structures, which are typically partitioned by objective (Allison et al., 2005).

2.4 Discussion and Conclusion

The MDO literature review has been carried out in order to find a better way to organise the complex engineering system optimisation design. MDO strategies provide a chance to explore the optimisation problem better by looking at the engineering system from a different point of view. Single level MDO strategies have been shown to have the capability of dealing with engineering systems having a small number of strongly coupled disciplines, especially two or three disciplines (Cramer et al., 1994 , Depince et al., 2007 , Kodiyalam and Center, 1998 , Sobieszczanski-Sobieski and Haftka, 1997). The AAO method has been shown to have the ability to handle different types of MDO problem and provide a solution. However, it results in heavier computation by exploring the data communication of those interdisciplinary coupling variables. MDF is one of the common ways of approaching the solution of MDO problems (Kodiyalam and Center, 1998). In MDF multidisciplinary feasibility is achieved by performing a set of analysis following the natural interdisciplinary coupling disciplines. Nevertheless, the optimiser of IDF has a fully control of all design variables that involve the interdisciplinary coupling variables.

Multi-level MDO strategies aim to distribute the difficulties of large scale (large number of design variables) optimisation problem, by organising the optimisation problem in different levels and decomposing the optimisation problem into a number of disciplines (Sobieszczanski-Sobieski and Haftka, 1997). The TCO Approach has been demonstrated in some implementation and shown the benefits of (Sobieszczanski-Sobieski et al., 1998) improving efficiency and exploring the design space better (Braun and Kroo, 1995 , Kroo, 2004). Sobieski introduced the BLISS optimisation approach to reduce the difficulty of the complex system optimisation by using the gradient-guided path to reach the improvement of system level design, which is gathered from sensitivity analysis (Sobieszczanski-Sobieski et al., 1998). However, the BLISS approach requires derivative information, which may not be readily available (Allison et al., 2009). Papalambros (2001) demonstrated the ATC MDO framework to solve hierarchical multi-level optimisation problems, which are typically partitioned by objective. Optimisation problems with TCO implementation are partitioned by constraints (Allison, 2004).

2.5 Summary

This chapter has briefly reviewed optimisation algorithms, from the basic concept of optimisation to the most advanced and up to date optimisation methodologies. An elaborate discussion of Evolutionary Algorithms was provided based upon GAs. The review of optimisation methodologies of Multidisciplinary Design Optimisation has been introduced – these help us to understand the higher-dimensional and complicated optimisation problem from a higher level, in order to broaden the mind of organising the complex optimisation problem more efficiently and explore the design space of large optimisation problem better.

Chapter 3 Literature Review: Diesel Engine Mapping and Calibration

3.1 Introduction

The purpose of this chapter is to review the relevant Diesel engine model based calibration literature in order to establish the 'state of the art'. A brief overview of Diesel engine technology is given to help with understanding of modern Diesel engine application. An overview of the Model Based Calibration process is presented, which involves engine Design of Experiments (DoE), engine response model fitting, and engine calibration optimisation. Different Diesel engine calibration processes are reviewed in detail and compared, in order to identify the Diesel engine calibration process issues. A discussion and analysis of the Diesel engine calibration process is presented.

3.2. Overview of Diesel Engine Technology

3.2.1 Emissions control for Diesel engines

Since the first firing of a Diesel engine on 10th August 1893 (with attendant black clouds from the exhaust pipe), the pollution problem has been a major consideration for the subsequent development of Diesel engines to the present day. Historically, developments were focused upon improving Diesel engine exhausts to be visually acceptable (Moon, 1974). In June 1927 the first production of Diesel trucks with acceptably clean exhausts were made by the Bosch Company. Clessie Cummins first experimented with truck Diesel engines in passenger cars in 1933, with the first installed Diesel engine in the Citroen Rosalie being sold in 1935 (Monaghan, 1998).

Many researchers have pointed out that the difficulty with Diesel pollution is to be found in the nature of the Diesel engine cycle, which relies upon auto-ignition when liquid fuel is sprayed into the combustion chamber late in the compression stroke. Figure 3.1 shows the theoretical pressure compression ignition process in a typical Diesel engine. From Figure 3.1, the injected fuel is vaporised and mixed with air during the period AB, termed the ignition delay. The pressure is increased rapidly by the uncontrolled combustion from B to C just before Top Dead Centre (TDC). From C to D, controlled combustion occurs at a rate determined by the air fuel mixture, and combustion from D is governed by the diffusion of the air fuel mixture (Stone, 1999).

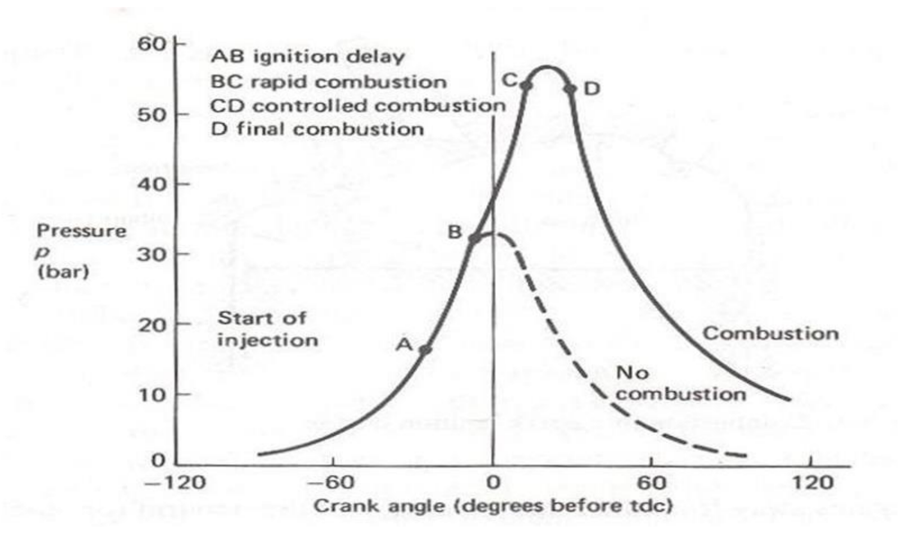


Figure 3.1 Hypothetical pressure diagram for a compression ignition engine (Stone, 1999)

Given that the air/fuel ratio in a Diesel engine is always lean, carbon monoxide emission is no longer a big issue for a correctly regulated Diesel engine (Stone, 1999).

Heywood (Heywood, 1988) summarised that most Diesel particulates are formed from incomplete combustion of fuel hydrocarbons. The incomplete combustion normally results from rich fuel injected in the diffusion-controlled combustion phase. Stone (1999) also discussed that fact that soot particles can be oxidised after the end of diffusion-combustion phase. The reasons for the increases in Diesel engine combustion smoke can be summarized as (Stone, 1999):

- The duration of diffusion combustion increases;
- The combustion temperature increases;
- Less oxidation of soot occurs during the expansion stroke since there is less time after the end of diffusion combustion and there is less oxygen.

Heywood (1988) explained that nitrous oxides (NOx) form at both the front and rear of flame gases. However, the combustion occurs at high pressure and the combustion timing is short. Since the Diesel cycle relies upon compression to achieve auto-ignition at high pressure, the combustion chamber reaches the reaction condition of NOx formation. Therefore, Stone and Ball (Stone and Ball, 2004) suggested reducing the NOx emissions for Diesel engine by two ways: reduce the compression ratio and reduce the combustion temperature.

Fuel economy is also an important criterion for judging an engine. Fuel economy depends strongly upon the quality of combustion. Heywood (1988) summarised a number of engine control parameters that affect the Diesel engine ignition quality, such as injection timing, fuel pressure, boost pressure, and Exhaust Gas Recirculation rate.

Combustion noise (which arises from the high pressure during the rapid combustion phase) is also a big concern for Diesel engine development, . As fuel is injected into the combustion chamber towards the end of the compression stroke, the fuel evaporates and mixes with air to form the flammable mixture. Nevertheless, the ignition does not occur immediately, while the flammable mixture is still being formed. During this delay the ignition occurs at many sites very rapidly. This rapid combustion produces the characteristic Diesel engine noise. Stone (2004) suggested two ways to reduce the Diesel combustion noise: reducing the initial rate injection or reducing the duration of combustion delay period. The duration of the combustion delay period can be reduced by increasing either the temperature or the pressure.

According to the requirements of reducing emissions and improving fuel economy, it is not too difficult to find that there are conflicts requiring complex trade-offs. Increasing the pressure can reduce the Diesel combustion delay, which leads to better Diesel combustion noise and fuel economy. However, this will increase NOx emissions (Stone and Ball, 2004). Additionally, an increase in temperature can reduce the ignition delay period and also increase the NOx emissions. Similarly, increasing injection timing delay leads to more fuel consumption and other trade-offs. Figure 3.2 shows the trade-off effect of injection timing and injection rate between NOx, smoke, BSFC, and combustion noise (Heywood, 1988).

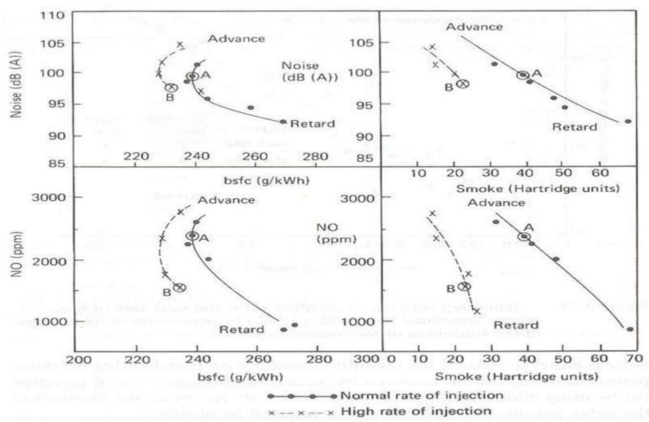


Figure 3.2 Trade-off between NOx, Noise, BSFC and Smoke for different injection rate and timing (Glikin, 1985)

Over the years several technical developments have been introduced into Diesel engines in order to satisfy emissions legislation and fuel economy requirements. The following sections discuss the most important technologies.

3.2.2 Turbo-charging Systems for Diesel Engines

The purpose of turbo-charging systems in Diesel engines is to improve the fuel economy as well as the combustion quality by increasing the density of the air boost into the engine (Stone and Ball, 2004). Given that modern engines use more technologies with complex controls, exploring the whole space of control parameters by using the traditional “one factor at a time” approach to calibration is clearly not feasible and very expensive. The more complete combustion improves the fuel economy and reduces smoke emissions. Almost all manufactured Diesel engines for ground vehicles are fitted with a turbo-charger (Tetsui and Ono, 1999).

Watson and Janota (1982) concluded that the turbo-charger has the ability to improve fuel economy. Figure 3.3 illustrates a comparison of the brake-specific fuel consumption of a turbo-charged engine against that of a naturally aspirated engine, scaled for the same maximum torque. This shows that the BSFC can be improved with higher boost pressure provided by a turbo-charged engine, especially at lower torque output (Stone and Ball, 2004).

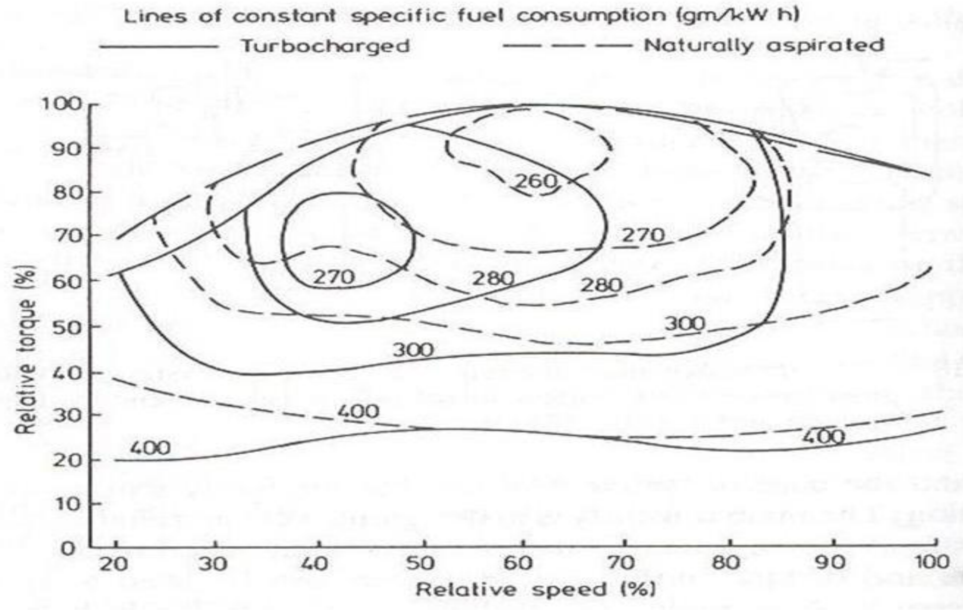


Figure 3.3 Comparison of BSFC between turbocharged and naturally aspirated engines with same maximum torque (Stone and Ball, 2004)

In addition, the higher pressure and temperature reduce ignition delay, which leads to better Diesel combustion noise (Stone and Ball, 2004). However, more NOx is produced with the higher pressure and temperature conditions.

3.2.3. Exhaust Gas Recirculation System

Exhaust Gas Recirculation (EGR) systems are used in modern Diesel engines to increase engine efficiency and reduce emissions (Lee, 2005). By using an EGR system, the oxygen and nitrogen in the air/fuel mixture are replaced by the exhaust gas (such as N2, O2, H2O, CO2, CO, particulates, and unburned HC) (Stone and Ball, 2004). Consequently, the emission of NO_x is reduced during the engine combustion process by the reduction of oxygen and nitrogen in the combustion chambers. The graphs in Figure 3.4 illustrate the relationship between NOx and EGR (Yokomura et al., 2004).

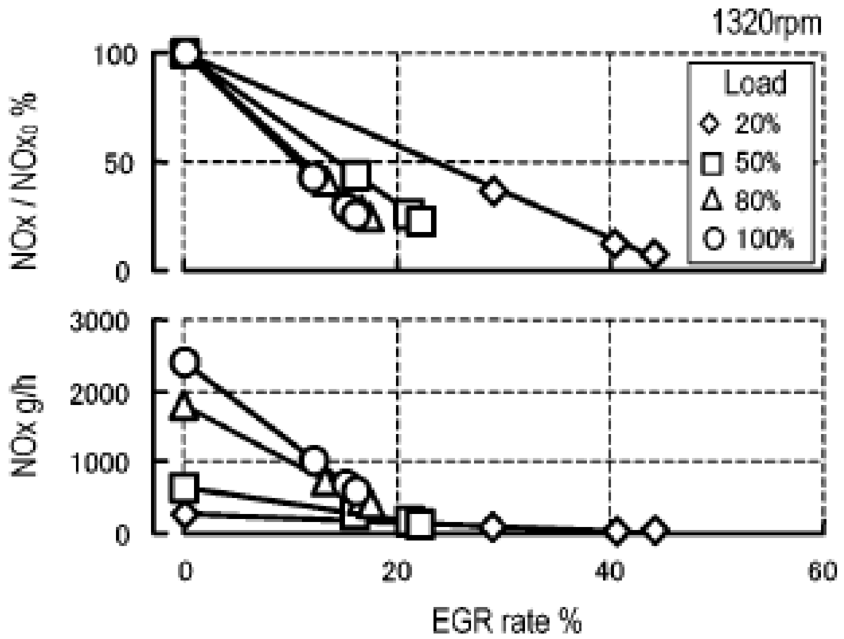


Figure 3.4 The Effect of EGR on NOx Emissions (Yokomura et al., 2004)

3.2.4 Common rail injection system

The electronically controlled common rail fuel injection system was introduced into modern high speed direct injection Diesel engines and permits a considerable improvement in fuel consumption and other engine performance (Stumpp and Ricco, 1996 , Boehner, 1997 , Stotz et al., 2000 , Guerrassi and Dupraz, 1998). The main difference compared to the traditional injection system is that injection pressure can be controlled at all engine operating conditions within the high pressure range (Guerrassi and Dupraz, 1998 , Huhtala and Vilenius, 2001). This also provides the potential for a multi-injection strategy. The high injection pressure can reduce the Diesel ignition delay, and also the multi-injection can reduce the injection rate by post injection. This will benefit the Diesel combustion noise, fuel economy, and smoke emissions. The fuel consumption and emissions can be varied by controlling the rail pressure (Park et al., 2004). The injection timing not only affects the fuel consumption but also affects emissions, torque and exhaust temperature. Park (2004) discussed the effects of fuel pressure and injection timing on the emissions and other engine performances, as shown in Figures 3.5 and 3.6.

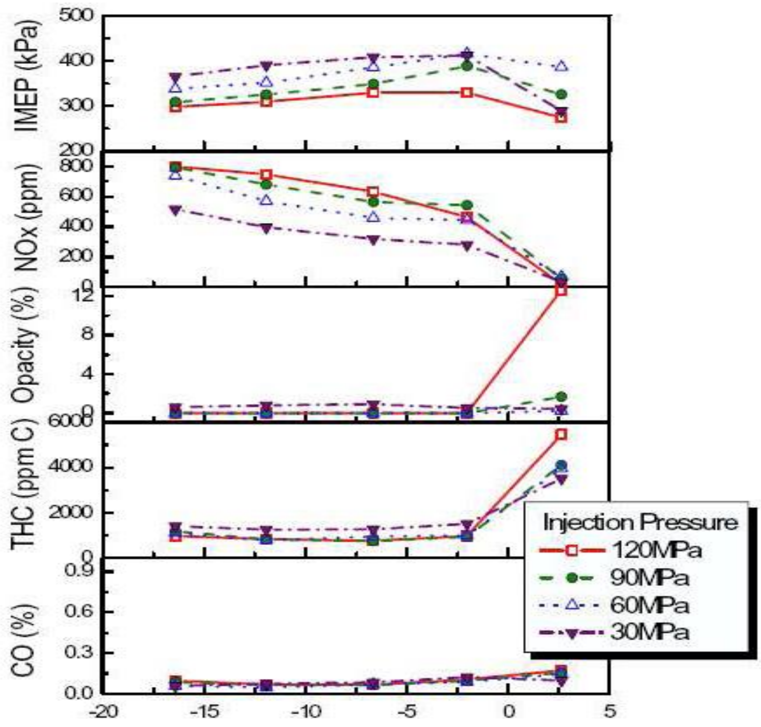


Figure 3.5 Effects on emissions of injection pressure and timing (ATDC) (Park et al., 2004)

Figure 3.5 shows that higher injection pressure results in lower emissions apart from NO_X; advancing injection timing increases CO, HC and Particulate Matter (PM) and decreases NO_X and Indicated Mean Effective Pressure (IMEP). Injection before TDC always benefits the engine performance, apart from NOx.

The electronically controlled high pressure common rail injection system also has the capacity to implement more than one discrete injection phase in a single cycle, i.e. main injection, pilot injection, and post injection. In the pilot and post injections, small amounts of fuel are injected at a specified crank angle before and after the main injection (Minami et al., 1995). The pilot and post injections respectively reduce the engine combustion noise and improve the catalyst efficiency (Guerrassi & Dupraz, 1998). Figure 3.6 shows the effects of pilot and post injections on engine performances Park (2004). With pilot injection, the main injection timing can be retarded without affecting ignition delay, which benefits NOx emissions and combustion noise. Also, Figure 3.6 shows that post injection results in a considerable reduction of emissions.

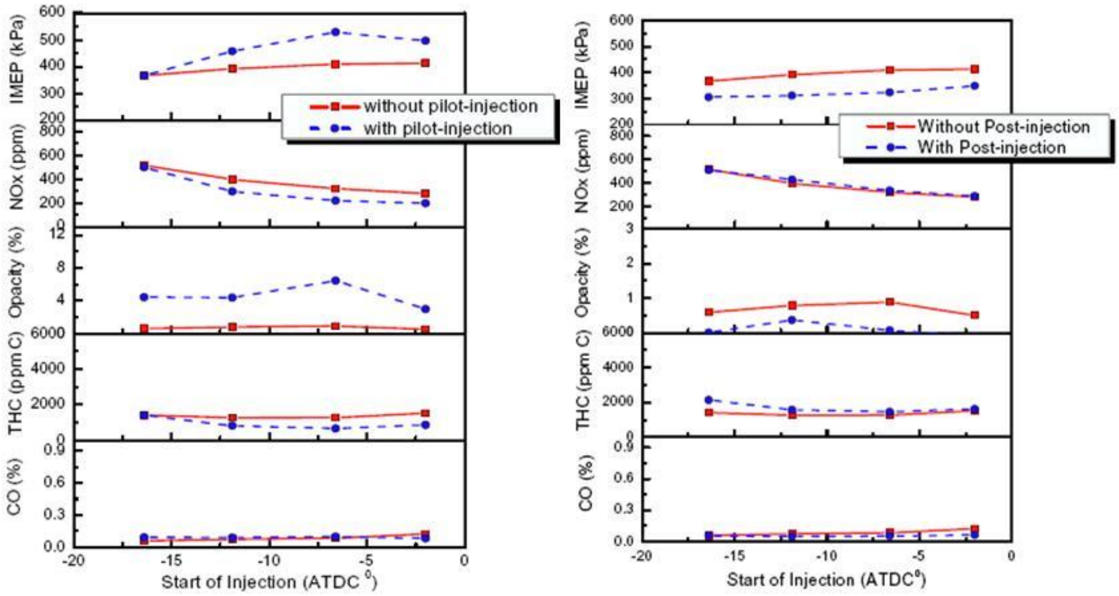


Figure 3.6 Effect of pilot and post injections on engine performance (Park et al., 2004)

3.2.5 Control Strategies for Diesel Engines

While the implementation of Diesel technologies helps with improving the engine performance, there is no single Diesel technology that can improve all aspects of engine performance, and thus trade-offs between different engine responses are required, as indicated previously. Therefore, a combination of technologies is required to improve engine performance.

Initial Diesel engine development was aimed at reducing emissions focused on reducing NO_x emissions by retarding the fuel injection timing (Baert et al., 1999). Smith and Tidmarsh (Smith et al., 1998) also carried out work showing that a combination of EGR and common rail injection strategy have the ability of tuning the emissions performance of a high speed direct injection Diesel engine. Figures 3.7, 3.8 and 3.9 show the trade-off between NOx and Particulates, BSFC and unburned HC by varying injection timing and holding the EGR rate at different levels. This shows that a significant effect on engine emissions and performance can be produced by tuning both EGR and injection timing.

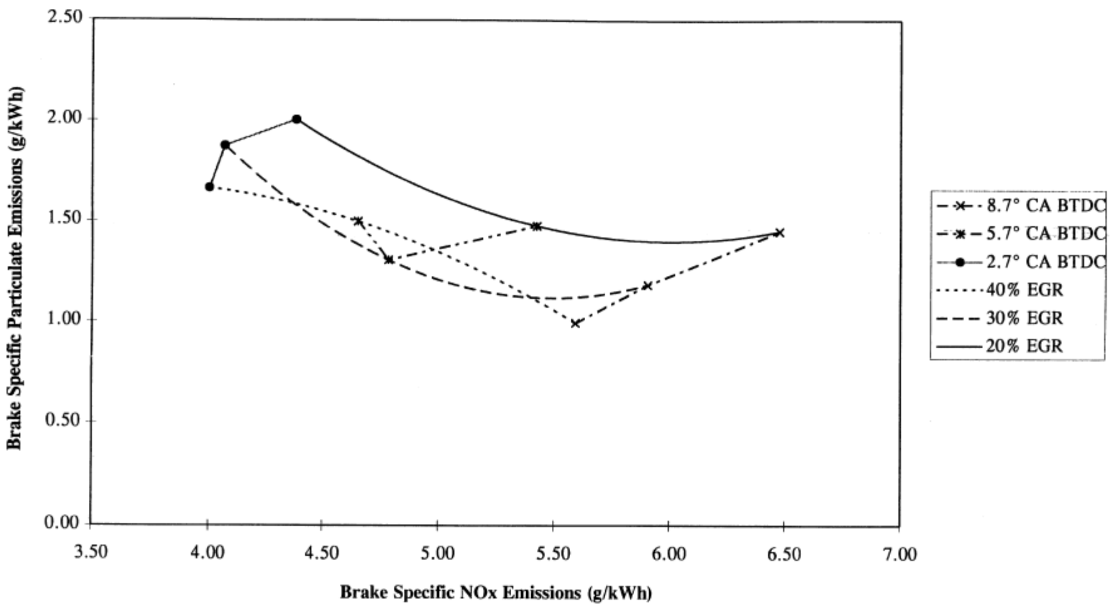


Figure 3.7 The Effect of Start of Injection Timing and EGR Rate on Particulate/NOx trade-off (Smith et al., 1998)

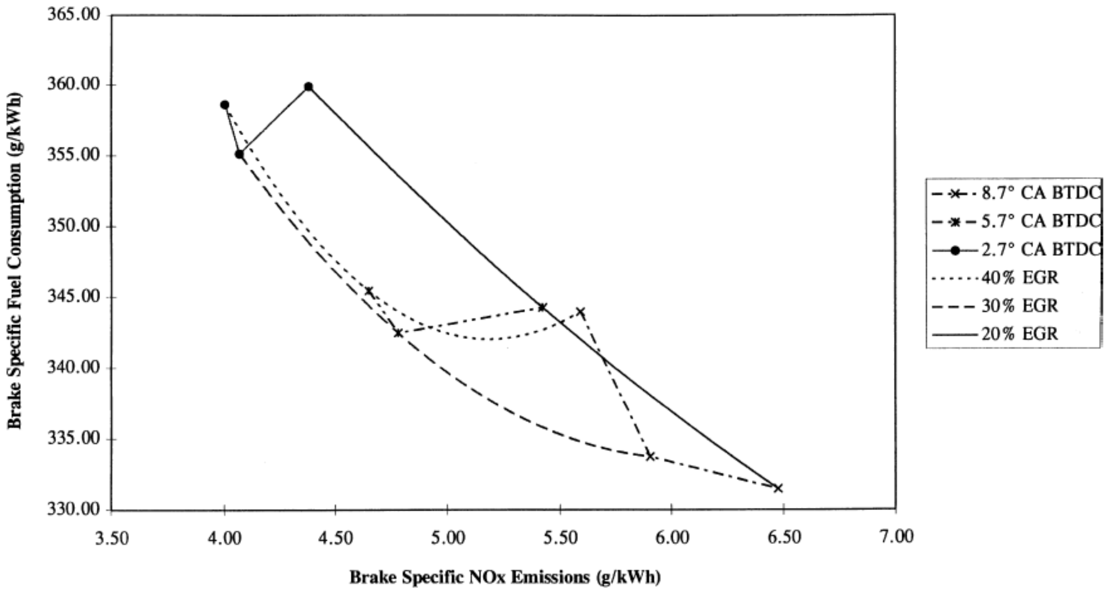


Figure 3.8 The Effect of Start of Injection Timing and EGR on BSFC/NOx Trade-off (Smith et al., 1998)

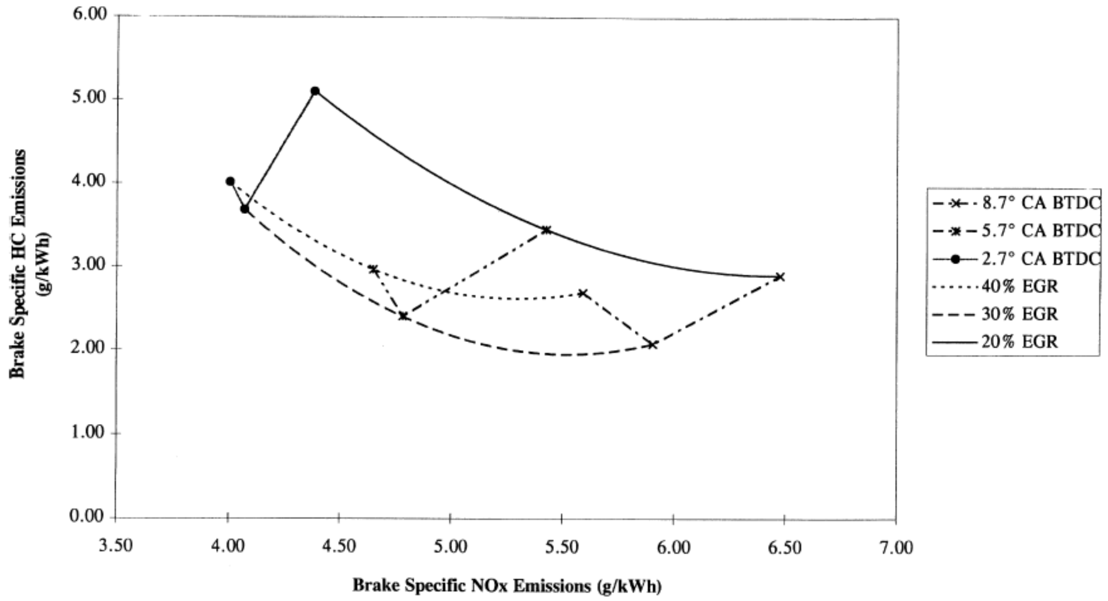


Figure 3.9 The Effect of Start of Injection Timing and EGR Rate on the BSHC/NOx Trade-off (Smith et al., 1998)

3.3. Model Based Calibration Process

3.3.1 Engine Control Unit Strategy

Diesel engines equipped with modern technologies (turbo-charger, EGR, and common rail injection system) are controlled by an electronic engine management system, the Engine Control Unit (ECU). The diagram in Figure 3.10 illustrates a closed loop engine control system strategy.

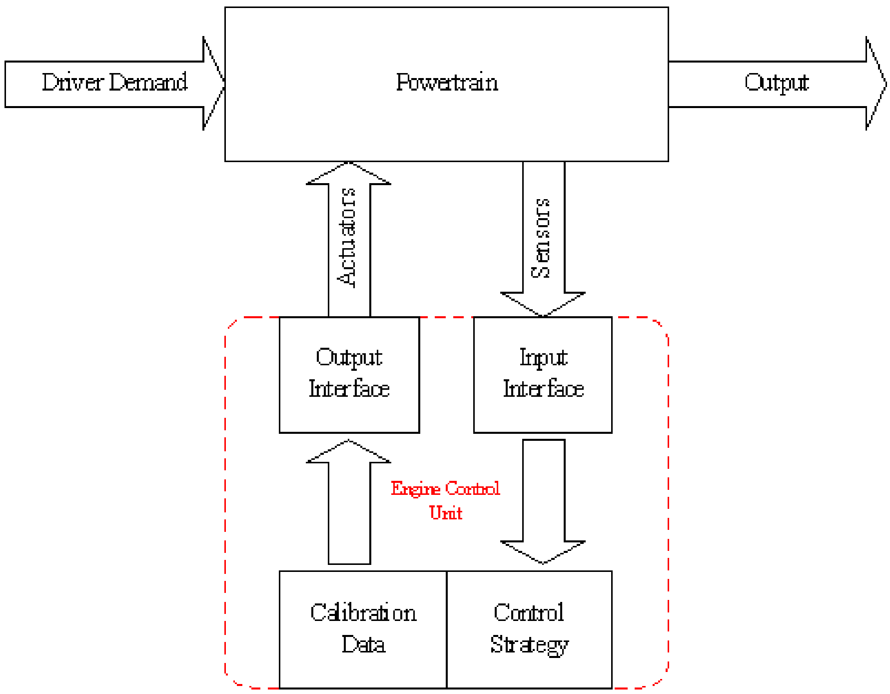


Figure 3.10 Illustration of Engine Control System (Cary, 2003)

The control strategy is executed in the form of a continuous loop by the microprocessor controller. The microprocessor receives signals from a range of sensor devices for engine status, such as engine speed, throttle position, Manifold Absolute Pressure, air temperature and engine temperature, and so on (Cary, 2003).

These signals are translated and interpreted as inputs. According to these inputs, the microprocessor determines the amount of actuation for the engine controls (actuators) corresponding to pre-stored calibration data. A look-up table was used in the Engine Control Unit, which is defined by engine actuator setting over the entire range of engine load/speed (Cary, 2003). In order to define a look-up table, the engine actuator settings are given at in a table known as "look-up table" which is defined by the engine load speed points over the engine operating range. On the other hand, the engine actuator settings can also be represented as functions of engine load/speed instead of the look-up table. Both the look-up table and the engine actuator setting functions are called 'engine operating map' that is produced by the engine model based calibration process (Goodman, 2006b).

3.3.2 Overview of Model Based Calibration Process

In general, engine mapping and calibration is the process of defining the engine calibration data, which is embedded into the ECU in order to control engine actuators. As a result of modern engines using more technologies, exploring the whole space of control parameters by using traditional one factor at a time experiment-based calibration process has became very expensive. Therefore, mathematical and statistical techniques are attractive for finding optimal solutions that produce low emissions and better engine performance, by using Design of Experiments (DoE), statistical modelling, and optimisation methods.

A typical model based engine calibration approach for steady-state engine tests is illustrated in Figure 3.11 (Sampson, 2010), and has been commonly used both for gasoline and Diesel engines, and can be summarised as follows:

1. **Design of Experiment (DoE):** is a more efficient way of exploring the design space by planning engine tests and aiming to reduce the cost of engine testing;
2. **Engine Test & Data Collection:** carry out the engine tests and collect the engine testing data according to the experiment design;
3. **Engine Responses Modelling:** based on the engine testing data, fit empirical mathematical models to predict the engine responses;
4. **Optimisation & Engine Control Map Calibration:** the engine response models are used for the optimisation to search for an optimal solution of the engine actuator settings; the aim is to fill the engine maps with local optimal engine actuator settings and to smooth the actuator transitions map between these settings;
5. **Implementation:** embed the calibrated engine control map into ECU and validate at system level (engine and vehicle installation).

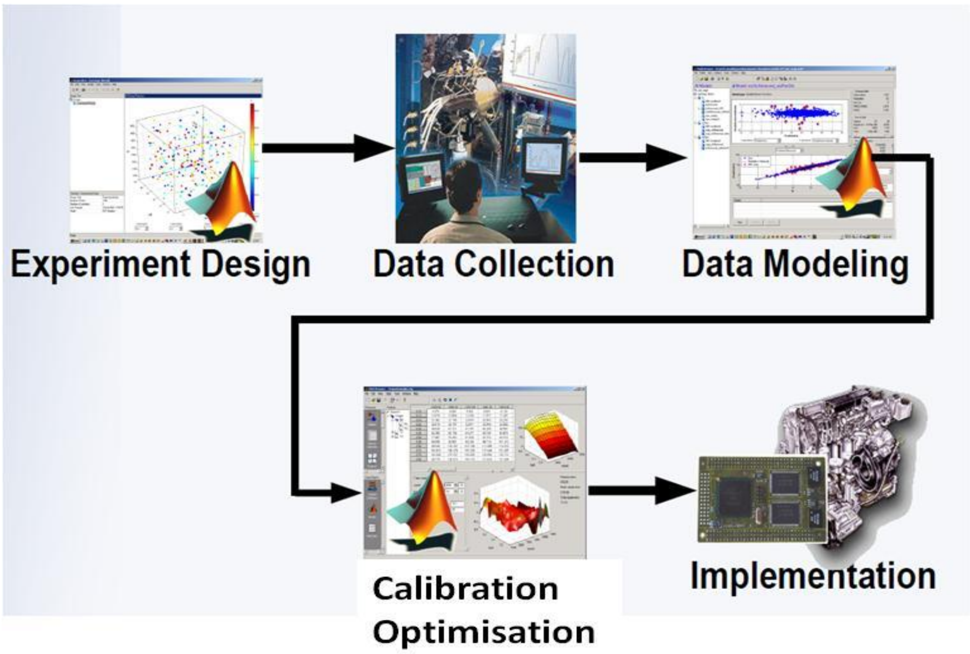


Figure 3.11 Illustration of Model Based Calibration Process (Sampson, 2010)

Steady-state model based calibration refers to the steady-state engine tests, which are to measure and collect engine response data while the engine is running at a constant engine load and speed condition. Therefore, this type of engine calibration approach is called steady-state model based calibration. Engine response models based on this type of engine test data can predict the engine response only at specified engine load-speed conditions. Such engine response models are commonly called “local models” (Roudenko et al., 2002). Optimisation with this type of engine model would normally be carried out individually at different engine load-speed conditions. Alternatively, the total objective formulation over the full range of engine load and speed conditions has to be integrated by summing all the weighted engine local models value (Hafner and Isermann, 2001 , Roudenko et al., 2002). Optimisation based on steady state engine models produces an optimal solution for a set of engine actuator settings for different engine load and speed conditions. With this set of optimal solutions, the engine control map is filled across the full range of engine load speed space. This steady-state model based engine calibration approach has been taken up by most commercial software tool packs, such as AVL CAMEO (Stuhler et al., 2002), MBC toolbox in MATLAB software environment (Styron, 2008) and so on. In the literature, a large number of applications of Diesel engine calibration are based on the steady-state engine calibration (Alonso et al., 2007 , Baert et al., 1999 , Brooks and Lumsden, 2005 , Haines et al., 2000 , Jankovic and Magner, 2004 , Jankovic and Magner, 2006 , Roudenko et al., 2002 , Sampson, 2004).

From another point of view, the steady-state engine calibration for a gasoline engine typically requires more than 30 different engine load/speed points in order to populate the look-up table over the range of engine load-speed (Sheridan, 2004). Comparing the gasoline and Diesel engine calibration processes, the gasoline engine calibration will require more expensive engine testing and more model fitting and optimisation to be carried out. In order to reduce the cost of engine testing and modelling, and to characterise the engine responses on the entire engine operating condition, a global engine modelling approach has been introduced (Alonso et al., 2007 , Atkinson et al., 2008 , Atkinson et al., 1998 , Seabrook, 2000). The global engine response model is a function of all engine control variables and parameters including engine load and speed. The idea of the global engine response modelling approach is to represent the engine behaviour over the whole engine operating range with a single global engine model rather than using a number of local engine models. Global engine response models can reduce engine test cost by producing a Design of Experiments (DoE), which involves engine load and speed as design variables. A global engine model has the ability of predicting the engine response across the full range of engine load speed space (Atkinson and Mott, 2005).

Research has been carried out to demonstrate and compare the use of local steady-state models and the use of global models (Hafner and Isermann, 2001 , Roudenko et al., 2002 , Stuhler et al., 2002 , Rask and Sellnau, 2004). To find the optimal solution of engine actuator settings, optimisation is carried out at each of the engine load/speed points according to the look-up table. The use of steady-state engine local models requires engine tests and statistical modelling at each of these engine load/speed points. Whereas, the use of global engine models reduces engine testing effort by carrying out DoE that involves all engine controls and engine parameters (speed and load). The optimisation problem at each of the engine load/speed points is formulated as a single optimisation problem, in Equation 3.1 (Hafner and Isermann, 2001):

Minimise: $F(x) = k_1 \frac{NOx(x)}{NOx_{lim}} + k_2 \frac{Opacity(x)}{Opacity_{lim}} + k_3 \frac{Fuel(x)}{Fuel_{exp}}$

Subject to: $$\begin{gathered} LB < x < UB \\ local\ engine\ response < Limits \end{gathered}$$

Equation3.1

F is the optimisation objective defined as a weighted sum of NOx, “Opacity” (a measure of smoke or particulates emissions), and “fuel” consumption. The control variables x need to be within the range defined by LB (Lower Bounds) and UB (Upper Bounds), and other engine responses are defined as nonlinear constraints within the user defined limit. At each of the engine load/speed points, the optimal solution of controls is generated and entered into the look-up table.

Vossoughi and Rezazadeh (Vossoughi and Rezazadeh, 2005 , Vossoughi and Rezazadeh, 2004) (2004; 2005) applied a similar process with the global engine models. In their research, the optimisation problem was formulated as a multi-objective problem locally at each engine load/speed point. The optimisation at each engine operating point is to minimise both fuel consumption and the weighted sum of emissions (e.g. NOx, Opacity, HC and CO). Evolutionary Genetic Algorithms (GAs) have been tested to solve the multi-objective optimisation problem. Wu et al (Wu et al., 2006) also introduced an engine calibration approach to maximise torque output and minimise the fuel consumption and NOx emissions. In the optimisation process, an attempt to produce a smooth engine control map has been made by adding a penalty function (related to engine stability conditions) to the objective function.

3.4. Engine Statistical Modelling

3.4.1. Introduction

Engine response modelling is essential in the model based engine calibration process. The aim of engine response modelling is to build the statistical models that can accurately represent the engine performance over the ranges of a number of independent factors (i.e. engine load/speed, injection timing, fuel pressure and so on) based on test data collected from engine experiments. Because of the reality of the engine testing environment, it is impossible to record accurately all the engine responses (torque, emissions and fuel consumption and so on) while the engine load/speed condition is being varied.

The use of statistical experimental design was proposed in the early 1970s to reduce the cost of engine experiments (Edwards, 2000). The design of engine experiments has to capture all the necessary information in the engine testing data through the engine response model fitting. On the other hand, the design of engine experiments depends upon the type of engine response model that will be fitted. Consequently, the statistical engine modelling process includes two strongly connected components, which are Design of Experiments (DoE) and statistical engine response model fitting.

3.4.2 Overview of DoE for Engine Testing

DoE is a statistical methodology to make an experimental plan, which aims to make engine tests more efficient without loss of significant information for an adequate engine response model to be created.

Factorial design is one of the basic DoE methods used This finds the change in response produced by changes in the levels of the factors and the interaction between the factors (Montgomery et al., 2001). Factorial designs based on polynomials were widely used for testing engines that did not have many control variables (Grove and Davis, 1992). As the number of control variables increased in engine systems, the number of design points has rapidly grown in the factorial design. For instance, with 6 factors each having two levels, $2^6 = 64$ runs are required for a complete factorial design. Fractional factorial designs can reduce the number of tests associated with higher order interactions. This requires prior knowledge to indicate that the higher interactions are not significant. However, in the engine mapping process there is only limited prior knowledge available. Seabrook (Seabrook, 2000) illustrated the use of a central composite design method to generate an experimental design for a small number of dimensions of engine controls. Figure 3.12 shows a central composite design obtained by adding the central star points to the factorial cube design.

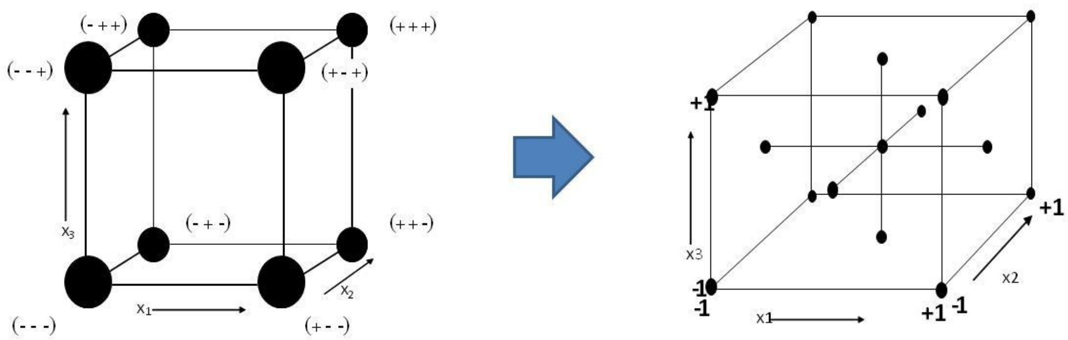


Figure 3.12 Composite design based on the two level factorial design(Montgomery et al., 2001)

However, as the number of engine control variables has increased rapidly and more complicated statistical models have been employed to better reflect the nonlinear behaviour of the engine responses, the requirements of engine testing data have dramatically increased. The standard DoE methods based on factorial experiments might not be appropriate for such a complicated design space (Cary, 2003 , Seabrook, 2000).

Space-filling designs are suitable when there is little knowledge of the engine behaviour within the operating envelope (Sampson, 2004). Space filling designs aim to provide a good scatter of data points in the space to be measured. One of space–filling design, Optimal Latin Hypercube (OLH), is a method for sampling a design space that uses an optimisation approach to optimise the uniformity of the distribution of a set of sample points. Several formulations for uniformity have been proposed to generate OLH design, such as maximising entropy, integrated Mean Squared Error (MSE) or maximising minimum distance between sampling points (Narayanan et al., 2007). Figure 3.13 shows the OLH design in a three dimensional space.

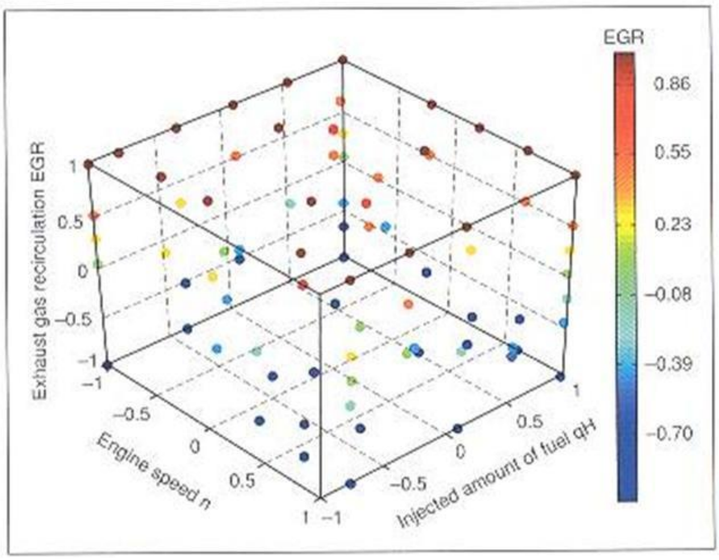


Figure 3. 13 Example of Optimal Latin Hyper Cube design in three dimensions space (Ropke, 2005)

Diesel engine techniques offer an opportunity to better control an engine. They also result in a more complicated design space. Chaudoye (Chaudoye, 2009) discussed the engine operating space varies for different engine operating conditions. Figure 3.14 illustrates the boost pressure and air mass flow limits of engine design space across a range of engine operating conditions.

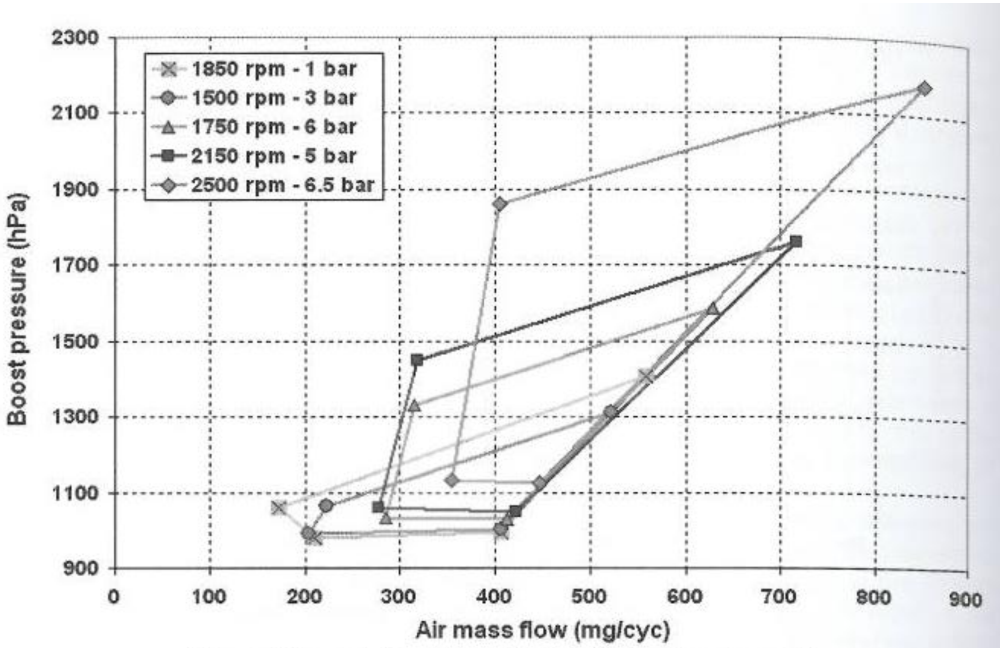


Figure 3.14 limited Engine Design Spaces (Boost Pressure Vs Air Mass Flow) over different engine operating conditions (Chaudoye, 2009)

Ward, Brace & Vaughan (Ward, 2004) suggested a study to determine the permissible envelope before undertaking the DoE task. Since the engine control parameter boundaries have been defined, the design space becomes even more complicated. Optimal designs give advantages if the behaviour (in the form of a suitable model type) is known from previous experience on similar engines (Sampson, 2004). Optimal design procedures assume that the functional form of the engine response models is known, and therefore the task is to select the design points according to a certain optimality criteria (Ayeb, 2005 , Ayeb et al., 2006). For example, the V-optimal design criterion aims to minimise the predictor error variance (Grove et al., 2004 , Jankovic and Magner, 2004), which is a common choice of optimal design in engine mapping.

3.4.3. Overview of Engine Response Modelling

Once engine test data is collected from an engine test bed, empirical models of engine responses can be built by using the response surface methodology based on the test data. A variety of types of models, both parametric and non-parametric, are commonly used for engine responses (Morton, 2002 , Ward et al., 2002). Polynomial models have been commonly used to develop "local" models for engine response features; they have been found to provide an adequate representation of engineering trends and a reasonably accurate approximation over relatively narrow operating spaces (Saunders, 2004). Non-parametric estimators can be useful when the responses of interest are relatively complex or there is little prior knowledge regarding the expected behaviour of the system. The Matlab® MBC toolbox provides a non-parametric modelling capability for Radial Basis Functions (RBF) models, which is a type of neural network (Morton, 2002). While the RBF models offer the attractive feature of local adaptability, they suffer from the curse of 'over fitting'. Research (Saunders, 2004) has shown that the choice of good measures for the quality of the model in a statistical sense (such as cross validation, Akaike Information Criteria or Bayesian Information Criteria) is very important. Such features are available in MBC and have been used in fitting models for the engine responses in this case study.

Since Holliday (Holliday, 1995) first suggested the use of a two-stage modelling method for Spark Ignition (SI) statistical engine modelling processes, this has become a widely used methodology for Gasoline engines.. The two-stage modelling matches the natural process of conducting gasoline engine experiments. The first stage concerns the modelling of engine responses (such as torque) over the spark sweeps, while the second stage is to fit a model of these engine responses or features as a function of other engine variables and parameters (such as engine load, speed and air/fuel ratio) (Davis and LAWRANCE, 2000). Table 3.1 shows engine input and output variables from one of the early two stage modelling applications (Davis and LAWRANCE, 2000).

Table 3. 1 Engine Mapping Inputs and Outputs

Inputs	Outputs
Air/Fuel Ratio	Torque
Exhaust Gas Recirculation	NOx
Spark Timing	CO
Speed	HC
Load	

The principle of two-stage modelling can be applied more generally for 'global' engine response models.

The introduction of the two-stage engine modelling approach (Holliday et al., 1998) for SI engine calibration required the introduction of more complex models capable of representing engine behaviour over the whole operating space. Similarly, the broad requirements for a 'global' model are to adequately represent the engineering trends and also to have flexibility or local adaptability to capture 'local' changes.

Polynomial models are generally not adequate for global models or when more variables are involved; higher order polynomials can model the general trends, but still lack the local flexibility. Alternative modelling strategies have been introduced, such as semi-parametric models based on cubic splines (Grove et al., 2004) and non-parametric models including artificial neural networks (Han et al., 2004), radial basis functions (Morton, 2002), and stochastic process models or Kriging (Jeong et al., 2006). Of these, radial basis functions and Kriging models are the most commonly used in engine modelling practice.

As one type of Artificial Neural Networks (ANN), Radial Basis Functions (RBF) are a powerful tool, which has proved useful for fitting data points with a large number of inputs (Morton, 2002). RBF models take a general form which is similar to the polynomial models (Equation 3.2):

$$f_i = K\left(\frac{\|x - c_i\|}{\sigma}\right) \qquad \textbf{\textit{Equation 3.2}}$$

x is the vector of inputs and c_i is the centre of the i^{th} radial basis function term. $\|x - c_i\|$ is give by the Euclidean distance between input and centre c, shown in Equation 3.3:

$$\|x - c_i\| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \cdots + (x_n - c_n)^2} \qquad \textbf{\textit{Equation 3.3}}$$

$\sigma > 0$ is the width of the radial basis function, and K is a strictly positive symmetric function called the 'kernel' with a unique maximum at its centre c_i. Examples of commonly used kernels include Gaussian, multi-quadric, inverse multi-quadric, thin-plate spline, and logistic functions.

ANN methods have been used to obtain a global engine model and are statistical tools of artificial intelligence, which are efficient in solving a wide range of problems including many engine calibration applications (Han et al., 2004 , Howlett et al., 1999). This approach has the ability of predicting the engine response models of factors and interactions between factors over the load-speed range of the engine drive cycle (Alonso et al., 2007 , Atkinson et al., 1998) .

Kriging can be regarded as a generalisation of RBFs, and it has been introduced as an interpolation technique based on a stochastic process model (Forrester et al., 2008). The predicted response at a point x0 is calculated as the weighted average of the neighbouring points, i.e.

$$\hat{y}_j(x_0) = \sum \lambda_i * y(x_i) + Z(x)$$

Equation 3.4

λ_i are the interpolation weights, and $Z(x)$ is the realisation of a Gaussian stochastic process with mean zero, variance v, and non-zero covariance (Seabrook, 2000). In simple kriging, the weights are derived based on the assumption that the mean and the covariance of y(x) are known, thus the kriging predictor minimises the variance of the prediction error.

One of the problems associated with using non-parametric models is that they suffer from the curse of over-fitting (Cary, 2003). Model selection and validation must be carefully conducted such that the model balances the error of approximation with the error due to random fluctuations. Model validation is usually based on the statistical analysis of residuals, and relies on measures such as Root Mean Squared Error (RMSE) of residuals and Predicted Residual Error Sum of Squares (PRESS). More advanced model selection and validation criteria include generalised cross-validation, Akaike information criteria (AIC) and Bayesian information criteria (BIC) (Saunders, 2004). The information based methods (AIC and BIC) introduce a penalty for the effective number of terms in the model, so they have the effect of reducing the modelling sequence (i.e. the number of terms in the model), thus effectively addressing over-fitting.

Within the engineering arena, non-parametric RBF and kriging models have been widely used in recent years, in particular in conjunction with experiments involving computer based models where a surrogate model is sought in order to address computational expense. Within this context, studies comparing RBF and kriging models have been reported (Chandrashekarappa and Duvigneau, 2007 , Costa et al., 1999 , Peter et al., 2007) pointing to the fact that in general kriging models tend to be more accurate.

3.5 Review of Diesel Engine Calibration

The local steady-state engine response model and global engine response model have been widely used in Diesel engine calibration approaches. In order to carry out the engine calibration task, a preliminary study is required to select a sample of engine operating points (Edwards, 2000 , Edwards et al., 1997). The weight for each operating point is usually determined based on the evaluated residency time over a specified engine cycle, in case studies presented by Edwards (1997; 2000) The engine tests were initially carried out allowing all the control factors (engine hardware design and control settings) to vary at each steady state engine operating point. Table 3.2 shows the engine input and output variables that are of interest.

Table 3.2 The Use of Diesel Engine Inputs and Outputs (Edwards et al., 1997)

Inputs	Outputs
Injection pressure (common rail fuel pressure)	Brake Specific Fuel Consumption (BSFC)
Initial injection rate (via a one-way orifice)	Brake Specific Particulates Matter (BSPMs)
Overall injection rate (via the nozzle hole area)	Brake Specific NOx
Number of nozzle holes	Brake Specific CO
Start of injection timing	Brake Specific HC

The “local” (i.e. at each engine speed/load point) optimisations were carried out to minimise BSPM first, then the optimal BSPM values were held as constraints to minimise the BSFC locally. The local optimal solutions were used to fill the engine control maps with some adjustment to smooth the engine map. In this case, the system was optimised for minimum fuel consumption; the PM limit was chosen to be slightly higher than the value achieved in the search for the minimum PM by consideration of the trade-off between NOx and PM (Edwards, et al, 1997).

Haines and Dicken (Haines et al., 2000) also introduced a similar process for Diesel engine calibration with steady state engine models. The steady state engine tests were carried out over 10 engine load/speed points and second order polynomial engine models were fitted. The estimated cycle fuel consumption was minimised, and estimated cycle emissions (NOx, CO, PM and NOx + HC) were set to be within the cycle emission constraints globally. Engine operating constraints were defined locally, such as smoke opacity, turbine inlet temperature, peak cylinder pressure and combustion noise. A smoothing procedure was used to obtain the smooth engine control map.

Roudenko (2002) developed an engine calibration approach based on the MOGA (Multi-Objective Genetic Algorithm) optimisation method. In Roudenko's study, the optimisation problem was formulated as minimising the fuel consumption and combustion noise with respect to EU emission legislation (Roudenko et al., 2002). Since more optimisation objectives were required, MOGA was used instead of a single objective optimisation approach to find the trade-off solution between fuel consumption and engine noise. In Roudenko's study, the optimisation problem has been formulated by two different approaches: global optimisation and point-by-point optimisation. These two approaches have been applied to the same engine and compared. The global optimisation used a number of models for different engine load/speed points to represent the engine performance over the entire engine operating range. The estimated total cycle fuel consumption was considered as the objective, while gaseous emissions were set as constraints. In point-by-point optimisation, the problem was formulated as minimising fuel consumption and combustion noise, while subject to the associated local emissions constraints at each engine operating point. The local emission constraints were from a preliminary study and were defined as the total cycle emissions limits divided by the weights. Roudenko (2002) concluded that "global" optimisation had produced a remarkably better solution than point-by-point optimisation. At the same time, point-by-point optimisation could not produce an optimal solution for every engine operating point.

Brooks et al (2005) presented another calibration approach based on steady state modelling. In this approach 14 model testing points were selected to cover a range of speed and load conditions over the New European Drive Cycle (NEDC) region of operation. The engine tests were carried out under steady state operating conditions. The models of response at each mode were generated by the Model-Based

Calibration (MBC) toolbox in the MATLAB software environment. Statistical analysis was applied to each testing point to find a basic trade-off between all the responses and the influence of individual factors and interactions between factors. Brooks (2005) illustrated that one of the trade-off solutions selected as an initial calibration point, while producing reasonable NO_x emission and Brake Specific Fuel Consumption (BSFC), was unacceptable from the point of view of particulate (smoke) emissions. Upon the statistical analysis, smoke emission was reduced by varying factors while subject to NO_x and BSFC target values that were defined by the basic calibration point. With such an approach, a trade-off between NO_x and smoke emissions is generated by a procedure of defining different NOx and BFSC values in the target range (Brooks, et al, 2005). The same process was repeated at five selected engine testing modes. From the same analysis at all these testing points, similar calibration changes are required in the region of NEDC. These have been extrapolated over the remaining NEDC region of the operating map creating an updated calibration (Brooks, et al, 2005).

Global engine models have been used in the engine modelling process to satisfy the requirement of reducing expensive engine test effort and the difficulties of optimisation by Atkinson (1997; 1998). As discussed in earlier sections, design of experiments techniques were required to be applied to the design space, which include engine load/speed space. The global engine models were fitted with Neural Network (NN) data-driven methods (Atkinson and Mott, 2005). A complete procedure for such a process has been given in the literature (Atkinson et al., 2008), described as follows:

- Carry out DoE (space filling design) for transient modelling over the drive cycle;

- Test and gather the designed operating points at steady state conditions;
- Carry out statistical data analysis on collected engine testing data;
- Apply neural network to generate global models over the range of engine operating conditions;
- Carry out local optimisation at each specified engine load and speed point, which are defined in engine look-up table. The global engine response models are used to define the objective functions and non-linear constraints functions by using equally load and speed constraints;
- Go back to the engine test bed and validate the optimal solution.

Han et al (2004) applied a neural network with a fuzzy logic system for engine calibration over the whole operating space. In this approach neural networks are built up over selected engine operating condition points. The local engine models were used to carry out optimisation to produce local optimal solutions for each of these the engine load/speed points. The local optimal solutions were expanded from the tested points to the whole range of engine operating conditions by the Adaptive Network-based Fuzzy Inference System (ANFIS). In the other word, ANFIS was used as an interpolation tool.

Alonso et al (2006) also introduced a Diesel engine calibration process based on an ANN modelling method with a GA optimisation approach. Because of the large number of parameters involved in the ANN model, it is not possible to carry out the optimisation of the whole range of engine operating conditions. The optimisation problem was simply split into a number of ‘local’ optimisations, which are defined by engine speed and fuel mass injected. At each of these ‘local’ optimisation problems, Brake Specific Fuel Consumption (BSFC) was optimised as an objective and two different types of constraint were defined, which were the limits of engine operating parameters and estimated emission constraints based on legal limitations of European emission test cycles. The engine operating parameters limits were defined as the range of minimum and maximum of the engine operating conditions. Emissions constraints were calculated from the legal limits and the approximate time, which spent on different engine operating condition ranges during the European emission test cycle. Similar methods have been widely used in literature (Omran et al., 2007 , Wu et al., 2006 , Zweiri and Seneviratne, 2007). For example, Omran's approach is illustrated in a flowchart in Figure 3.15 (Omran et al., 2007).

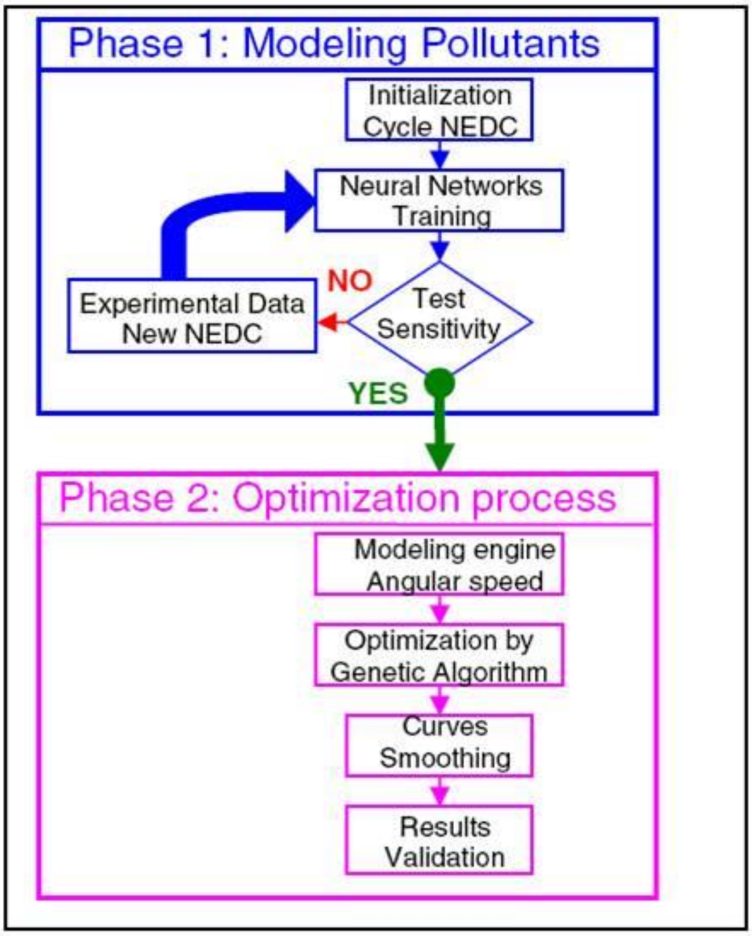


Figure 3.15 A Diesel Engine Calibration Approach with ANN Engine Global Models (Omran et al., 2007)

Desantes (Desantes et al., 2005) introduced a similar calibration approach based on the applied ANN method. In this approach, 7 modes of steady state engine operating points were chosen with 7 input variables at each node. The ANN models were used to specify BSFC as an objective and emissions as constraints. A Sequential Quadratic Programming (SQP) optimisation method was implemented for the weighted aggregation of the 7 modes of the different engine operating conditions. The optimal solutions for the different engine operating conditions are integrated into an engine map covering the whole engine drive cycle by interpolation.

Using dynamic engine testing techniques , there are more options for fitting a global engine response model over the entire operating range. The typical global dynamic engine model is described as follows (Neßler, 2007)

$$F(t) = f\big(X(t), X(t-1), \ldots X(t-k), F(t), F(t-1), \ldots, F(t-l)\big)$$

Equation 3.4

X is a vector consisting of engine control variables and parameters (i.e. engine load/ speed, fuel pressure, injection timing and so on), t is the present time, F is the engine response to be modelled, and k and l are the delay and response time of the instantaneous measuring device, respectively. Accordingly, the optimisation problem can be formulated as Equation 3.5:

Minimise: $f_i(t) = \int_0^T F_i(t) \cdot dt$

Subject to: $P_j^{lower} < P_j(t) < P_j^{upper}$

$g_k(t) = \int_0^T F_k(t) \cdot dt < lim$

$N(t) = N_{target}$

$L(t) = L_{target}$

Equation 3.5

To reduce the computational load of this optimisation problem, the optimisation is normally carried out at a number of engine operating points (Vossoughi and Rezazadeh, 2005). Therefore, the optimisation process based on the steady state engine model is still commonly used for dynamic model based calibration processes.

3.6 Summary

In order to satisfy emissions legislation and customer expectations, the engine industry has introduced strategies, such as common rail injection systems, turbo-charging systems and EGR systems. These systems provide several potential controllable actuator variables to produce better engine output performance. To get the better Diesel engine performance without the expense of other output performance, it requires a careful combination of such strategies.

The Modern Diesel engine calibration approaches focus on the efficiency of modelling and optimisation approaches, which can reduce fuel consumption and emissions with less cost of computation and engine testing. Also, producing a smooth engine map of optimal solutions over the engine drive cycle is another goal of engine calibration approach.

During the development of Diesel engine calibration, the difference of combustion between Diesel and gasoline engine suggest that a Diesel engine calibration process cannot take the same process as for a gasoline engine. A number of similar approaches have been suggested in the literature review. Table 3.3 summarises the literature on Diesel engine response modelling and calibration optimisation.

Table 3.3 Summary of Literature on Diesel Engine Calibration Processes

Reference	Used engine response model	Optimisation process
Edwards (1997)	Local engine response steady state models are used at 11 engine operating points	Simply optimized the engine response at each of selected engine operating point, and used these local optimal solution to fulfill (calibrate) the engine map
Haines, Dicken (2000)	Local engine response steady state models are used at 10 engine operating points	A similar process to Edwards' was suggested by carrying out the optimisation locally with a smoothing procedural to fulfill the engine map
Roudenko (2002)	Local engine response steady state models are used at 13 engine operating points	Compared between local optimisation and global optimisation (weighted sum of local response estimations) strategies, and claimed much benefit from global optimisation strategy.
Atkinson (1998, 2005 and 2008).	Global engine response models are used	Based on global engine response models (neural network), the optimisation was carried out across the Federal Testing Procedural, while subject to the emission constraints
Han et al. (2004)	Global engine response models are used	Used the fuzzy logic to train global engine response model, the optimisation were carried out at a number of specified engine load speed points and these local optimal solutions were used to fulfill the engine map by Adaptive Network-based Fuzzy Inference System (ANFIS)
Brooks et al (2005)	Local engine response steady state models are used at 14 engine operating points	The calibration approach was based on the steady state engine test at 14 engine load/speed points, and the engine calibration optimisation process was individually carried out at each engine load speed point
Alonso et al (2006)	Global engine response models are used	The optimisation was carried out by using the global engine response models while subject to different set of engine load speed constraints

From the literature reviewed it was found that two different optimisation formulations have been compared and can be summarised as:

- Both 'local' optimisation and 'global' optimisation formulations require preliminary studies of the engine's physical (hard) limits and the range of engine operating conditions.
- The emissions limits are defined as the emissions total mass flow based on the drive cycle, and need to be redistributed into each of the engine operating points according to the preliminary study.
- The redistributed emission constraints for local optimisation at each engine operation point potentially restrict the optimisation process for global optimisation (Roudenko et al., 2002). Correspondingly, global optimisation provides more flexible emissions limits or cost function targets between the engine operating points.
- In filling the engine control maps with the local optimal solutions, a smoothing process is required to ensure a consistency of engine operating conditions, so that it may reject the local optimal solution or make adjustment of the optimal engine actuator settings.

Most Diesel engine calibration approaches have been developed by advanced modelling methods and evolutionary algorithm optimisation. Advanced modelling methods provide a more efficient modelling process and accurate engine performance prediction. Evolutionary algorithms have the capability of handling the multi-objective optimisation problem. However, all the optimisation approaches are carried out at different engine operating conditions locally with the control factor changing limitation constraints, which narrows down the design space. Then optimal solutions are interpolated to the remaining engine operating conditions of the NEDC. Such a process might result in the optimal solution with control factors lying on bounds which are not robust or produce unsmooth engine maps and engine responses.

Chapter 4 Critical Review of Current Process for Diesel Engine Steady State Calibration Process

4.1 Introduction

The work in this thesis concentrates on a steady state model-based calibration optimisation for Diesel engines.

The aim of this chapter is to introduce a motivating reference supplied by the Sponsoring Company, which will be used as a basis for the analysis, development, testing and validation of the novel approaches to calibration optimisation problem formulation developed and implemented within this thesis. This reference case study is a V6 Twin Turbo 2.7L Diesel engine, currently in production at the Sponsoring Company.

The results obtained from the new methods developed in this work will be contrasted with those achieved by using a conventional, “current”, steady state Diesel calibration process used by the Sponsoring Company, which will be taken as a benchmark. Therefore, it is important to first critically review this process, within the broader context of Diesel engine calibration optimisation methods reviewed in literature, as the basis for establishing the methodology for further analysis and development.

Specific engine Case Study data and information will be provided along with the critical review of the “current” steady state calibration development process.

4.2 Critical Review of the Current Process

Figure 4.1 illustrates the Diesel engine steady state model-based mapping and calibration process used by the Sponsoring Company. In line with other Diesel engine mapping and calibration processes presented in the literature (Styron, 2008) and widely used by other companies, steady state engine testing is carried out at a number of discrete operating points in the engine speed – engine load space, referred to as “minimap” or nodal points. Engine response models developed for each of the minimap points are used to generate the “local” calibrations, i.e. optimal “local” actuator settings. A smooth “global” engine operating map (i.e. smooth actuator map) is then produced based on the local calibrations, by interpolation across the engine load-speed range.

A more detailed discussion of each of the process steps illustrated in Figure 4.1 is given in the following sections.

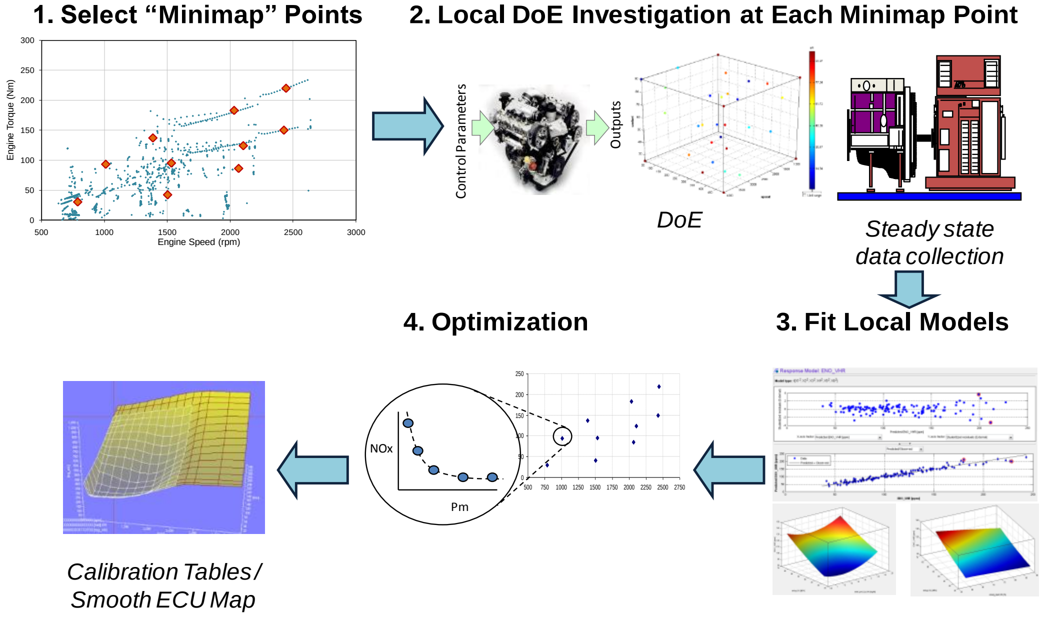


Figure 4.1 Diesel Engine Model Based Calibration Process (Yin et al, 2009)

4.2.1 Selection of Engine Testing Points

The first step in the process is to select representative points in the engine operating space (engine speed–load) at which engine testing will be conducted and minimap models developed.

In determining the number of test points there is a clear trade-off between effort (in terms of time and cost associated with engine testing) and accuracy, in relation to the “global” calibration (actuator maps across the engine speed – load operational range) which has to deliver good fuel economy and emissions across the driving cycle. For the engine case study considered in this work, 10 minimap points have been selected. This is consistent with similar case studies presented in literature, such as (Haines et al., 2000 , Styron, 2008).

The focus of the calibration effort is ultimately to ensure that the vehicle will achieve minimum fuel consumption whilst meeting constraints on emissions imposed by legislation. To measure the vehicle emissions, a reference emissions drive cycle was defined to simulate a typical driving profile, which consists of a series of accelerations, decelerations, and frequent stops (Andre et al., 1995 , Tzirakis et al., 2006). Across different countries, various testing criteria are used by different vehicle certification and regulatory authorities which are mainly based on two types of methodologies. One methodology is reflected by the Economic Commission for Europe (ECE) and Japanese cycles, that is made up of a series of repetitions of various vehicle operating conditions representative of typical driving modes (Kageson, 1998). The other methodology is the Federal Test Procedure (FTP) mainly used in the U.S.A, Canada, and Australia. The FTP cycle consists of 23 cycle tests in order to represent different modes of driving (Samuel et al., 2002). The product for the global market has to satisfy all the emission standards.

The New European Drive Cycle (NEDC) was introduced in the early 1990s (Kageson, 1998). NEDC is defined by a vehicle speed versus time sequence developed for presenting a certain vehicle driving pattern in a particular environment. The purpose of NEDC is to measure and regulate the exhaust gas emissions and monitor fuel consumption, where the emissions have to be within legal limits. Figure 4.2 shows that NEDC involves four segments of ECE-15 (known as Urban Driving Cycle, UDC) and the Extra Urban Driving Cycle (EUDC). The total duration of NEDC is 1186 seconds.

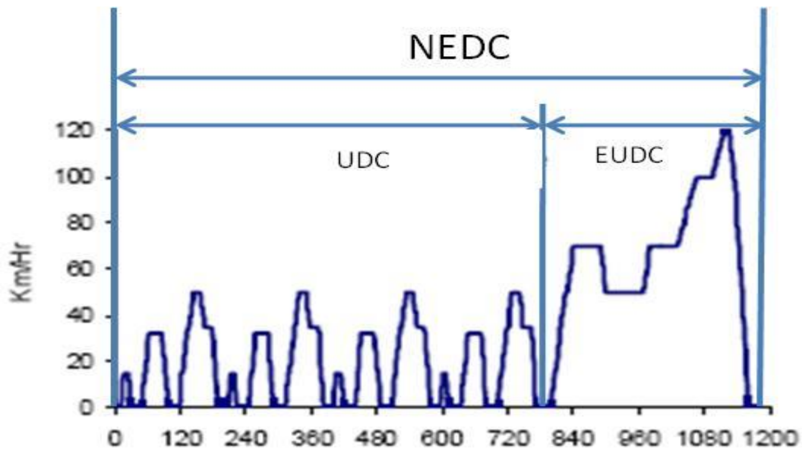


Figure 4.2 Illustration of NEDC Emission Test Cycle (Kageson, 1998)

Early in an engine development programme when the first steady state calibration is carried out, a vehicle simulation model is typically used to predict the engine speed/load operating conditions over the emissions drive cycle. This analysis generates a cloud of points as shown in Figure 4.3 that represent the second-by-second predicted engine speed-load operating points over the New European Drive Cycle (NEDC). If an engine is used in multiple vehicle applications, as it is the case at the Sponsoring Company, then several simulations will be conducted, resulting in a bigger “cloud” of points, which aggregates the engine residency across the range of vehicle applications considered.

In order to select a set of 10 minimap points that are the most representative of the whole drive cycle, an optimisation procedure is used (Goodman, 2006b). This process is based upon the minimisation of the sum of distances between each of the points in the cloud and the nearest minimap point. The selected minimap points are also shown in Figure 4.3.

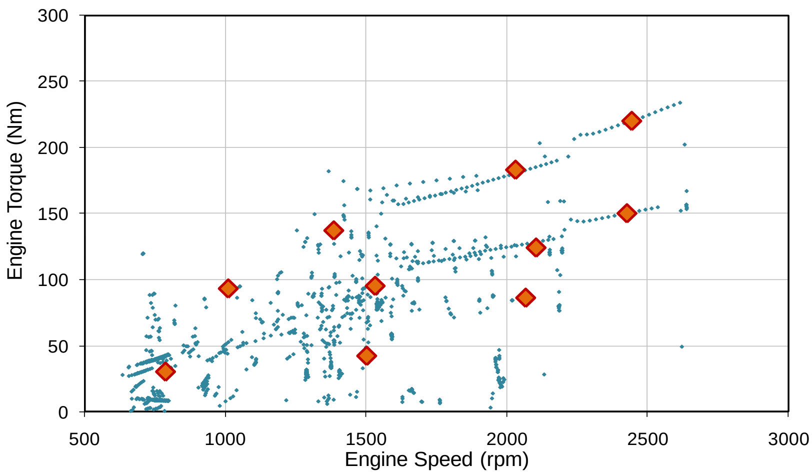


Figure 4.3 Illustration of Minimap Points Selection (Goodman, 2006a)

An equivalent residency time for each minimap point can be calculated based on the time the engine spends at the residency points associated with the respective minimap point (i.e. each “point” in the cloud, which represents 1 second in the engine operation during the NEDC cycle, is added to the nearest minimap point area). By normalising with respect to the total duration of the NEDC cycle (i.e. 1186 s) we can derive the equivalent “weight” of each of these minimap points in the cycle output in terms of fuel consumption and gaseous emissions. Table 4.1 illustrates the weight of the minimap points for a particular engine application (luxury saloon). Since the particular engine is used for different vehicles, the minimap points are selected from the combined drive cycle simulation of two different vehicle applications. Note that in Table 4.1, a zero weight is assigned for minimap point 8. This is due to the fact that this is a high load and speed point, which is used by other vehicle applications. Since Table 4.1 is only representing the ‘hot’ cycle (i.e. the engine temperature over the certain value), the sum of weight is 92.4%.

Table 4. 1 Equivalent Weight of the Selected Minimap Point for the Selected Engine / Vehicle Application

Minimap Points	Engine Speed (RPM)	Engine Load (Nm)	Weight (%)
1	750	30	41.6%
2	1010	93	1.3%
3	1530	95	9.4%
4	2065	86	2.8%
5	2029	183	0.3%
6	1383	137	1.4%
7	1505	42	30.4%
8	2442	220	0.0%
9	2102	124	4.0%
10	2425	150	1.4%
		Total	92.4%

4.2.2 Local DoE Investigation at Each Minimap Point

The next step in the process is to plan and conduct the steady-state engine tests. Besides engine speed and engine load, temperature is another global variable that plays an important role in calibration, in particular in relation to emissions. The common practice in steady state calibration testing is to “block” this variable by conducting testing at “hot” and “cold” engine temperature (usually defined in relation to the engine coolant temperature). The discussion and analysis below, and in this work, relates to “hot" calibration of the Diesel engine case study. Therefore, Table 4.1 only represents equivalent weights of residency for each selected minimap point from the ‘hot’ engine operating condition.

4.2.2.1 Analysis of Calibration Variables

In order to plan and conduct the steady state engine testing, the design space (engine control variables) and the decision space (engine responses of interest) must be analysed and defined.

The engine responses of interest include fuel flow (required to achieve the required torque) and gaseous emissions as engine outputs, such as Particulates Matter (PM)/ soot, HC, CO and NOx. These can be regarded as “global” responses, in the sense that the engineering (and legal) interest is on the cumulative values over the drive cycle rather than at a particular engine operating point. In addition to these global responses, calibration engineers are interested in other ‘local’ engine operating parameters such as engine noise (important to the customer), exhaust gas temperature and Indicated Mean Effective Pressure (IMEP) that are important to maintain the efficiency of catalyst, and combustion stability.

The engine control parameters or variables, which relate to engine calibration actuators, are usually defined by the calibration engineers and also depend on the ECU control / strategy. For the Diesel engine used as a case study in this work, the engine control variables included:

- Fuel Injection pressure/ rail pressure, which is controlled by control valve;

- Fuelling strategy parameters: Main Injection timing (start of injection), is given as the angles Before Top Dead Centre (BTDC) and accurately controlled by electronic control injectors;

- Multiple fuelling: Pilot injection – which involve quantity of fuel in the pilot and pilot timing (relative to the main injection timing);

- Mass Air Flow, which is measured by throttle position or air flow sensor;

- Variable nozzle of turbine demand position, which is described as Turbine nozzle position or valve position.

Table 4.2 specifies the engine control actuators with given engineering coded names and units.

Table 4.1 Engine Inputs Variables (Actuators)

Engine Inputs (Actuators)	Engineering Coded Name	Engineering Unit
Fuel pressure	Fup	MPa
Main injection timing (start of injection)	soi_main	degATDC
Pilot injection timing relative to main injection (difference to main injection)	soi_prev	degATDC
Fuel mass injected in pilot injection	mf_prev	mg/stroke
Mass air flow	maf	mg/stroke
Variable nozzle of turbine demand position	bpapwm	%

Figure 4.4 summarises the analysis of the engine control variables and responses for the Diesel engine case study.

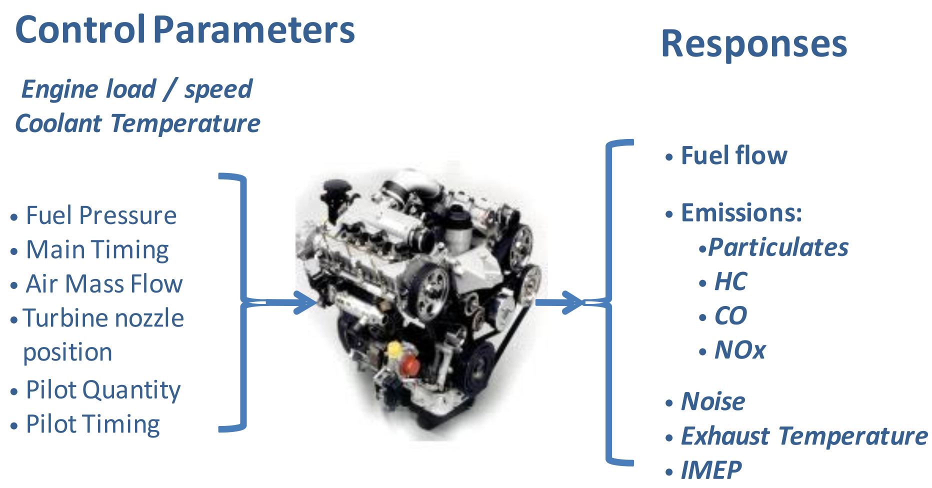


Figure 4.4 Analysis of Engine Control Parameters and Responses

Table 4.3 defines the engine response of interests with engineering coded name and unit.

Table 4.2 Engine Outputs (Responses)

Engine Response of Interest	Engineering Coded name	Unit
Indicated Mean Effective Pressure	IMEP	bar
Overall Apollo noise level	Apollo	dBA
Exhaust gas NO_x concentration	NO	ppm
Exhaust gas HC concentration	HC	ppm
Exhaust gas CO concentration	CO	ppm
AVL smoke meter Filter Smoke Number (FSN)	Smoke1FSN mean	FSN
Total fuel consumption	Emmf_tot	Mg/stk
EGR valve position	Emang_EGR	%

For each minimap point the design space is specified in terms of the actuator valid ranges, which are defined by the calibration team based on experience and preliminary experiments. Table 4.4 shows the valid engine actuator limits defined for each minimap / engine operating point of interest.

Table 4.3 Engine Actuator Limits at Different Engine Operating Points

Minimap Points	Operating condition Engine Speed (RPM)	Engine Load (Nm)	fup MIN	fup MAX	soi_main1 MIN	soi_main1 MAX	mfprev2 MIN	mfprev2 MAX	soiprev2DIFF MIN	soiprev2DIFF MAX	maf MIN	maf MAX	bpa MIN	bpa MAX
1	750	30	20	26	-5	5	0.8	2.2	10	20	210	260	60	95
2	1010	93	26	40	-3.2	5.3	0.8	2.4	5	25	270	370	60	95
3	1530	95	38.5	65	-2.5	5.44	0.8	2.5	9	29	280	330	60	90
4	2065	86	45	70	-4.5	3.5	0.8	2.4	8	35	290	360	50	80
5	2029	183	48	74	-5	2.25	1	3	12	30	465	565	45	70
6	1383	137	35	70	-3	4.5	0.8	3	10	35	350	460	60	88
7	1505	42	36	62.5	-1	4.88	0.8	3	9	29	215	300	70	92
8	2442	220	68	92	-8	-1.5	0.8	3.5	12	40	615	720	44	80
9	2102	124	54	80	-3.5	3.07	0.8	2.8	10.8	31	380	465	45	75
10	2425	150	57	85	-5.5	2	0.8	3	12	32	455	545	38	63

4.2.2.2 DoE Engine Test Plan

The engine tests are planned using a DoE approach. Space filling DoEs have been growing in popularity in engine mapping testing, in particular since software to generate space filling DoEs is widely available, including the Matlab Model Based Calibration toolbox. While this does not give any statistical advantage over more conventional DoEs (such as fractional factorials or Central Composite / Box Behnken DoEs) if a polynomial model is to be fitted (which has been common with Diesel engine response models), they do have practical advantages that (i) enables engineers to choose the number of tests they want to run, and (ii) the fitting of a more flexible model, such as radial basis function or Kriging can be attempted. The number of tests run in each experiment is based on a trade-off between effort (time and cost of running the engine tests) and the desired accuracy of the model.

For the engine case study, a space filling Optimal Latin Hypercube (OLH) design was chosen, and generated using the Model Browser of the Matlab ® MBC (Model Based Calibration) toolbox. The number of tests in the OLH DoE for each minimap point was 135, with further 20 tests planned as validation points.

At each test according to the DoE, engine speed and actuator settings are held as predefined constants from the DoE and the quantity of fuel injected from main injection is varied in order to achieve the specified torque from the DoE. The same procedure is carried out for all the tests according to the DoE. Engine actuator settings and engine outputs for each of test are recorded and collected. For each minimap point, a small number of extra tests were planned and collected as external validation data. For this experimental plan and for each minimap point, data was collected from an engine dynamometer testing facility over a period of 2-3 days.

Engine testing data was recorded into a data logger, which is an integral part of an engine testing bed facility. The engine test data was provided in Excel software format for each minimap point; as an example the engine test data for the minimap point 750rpm-30Nm is shown in APPENDIX I. This engine test data from the specified Diesel engine is used throughout this research.

4.2.3 Fitting Local Engine Response Model

Once the engine test data has been collected (i.e. measured engine responses obtained at each testing point in the DoE), response models can be fitted for each engine response and at each minimap point. Historically, third order (cubic) polynomials have been used for local Diesel engine models (Goodman, 2006b). This was also the choice of the calibration team at the Sponsoring Company, thus cubic models have been fitted at all minimap points for all engine response interests, shown in Table 4.3.

All models have been fitted using the Model Browser tool in the MBC Toolbox. PRESS RMSE was used as the statistical model fitting criteria (internal validation). External model validation criteria, in addition to residuals analysis, included RMSE for the validation set (points not tested in the main DoE) and engineering trends analysis were conducted. For reference, the fitted models are presented as an MBC file in Appendix E-1(Appendix in electronic format).

During the validation process for minimap point 1 it was noticed that the quality of the engine response models (across all responses) was worse than the engine response models at other minimap points, based on the analysis of model building PRESS and validation RMSE. After further analysis of the test data and discussion with the calibration team at the Sponsoring Company, it was realised that during the engine testing it was difficult to maintain constant coolant temperature at this minimap point (750rpm & 30 Nm). Response models were refitted for this minimap point, with coolant temperature as an explanatory variable, which considerably improved the quality of the models. Figure 4.5 illustrates a comparison of the NOx emissions response model with and without the engine coolant temperature; in terms of the residual plots. This shows a much smoother residual plot for the NOx response model fitted with coolant temperature as a variable.

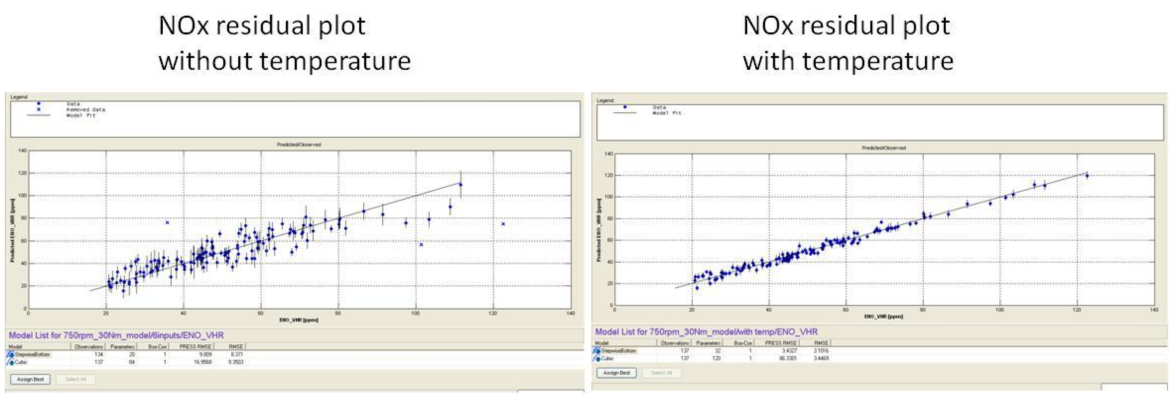


Figure 4.5 Comparative Residual Plot for NOx Models at Minimap Point 1

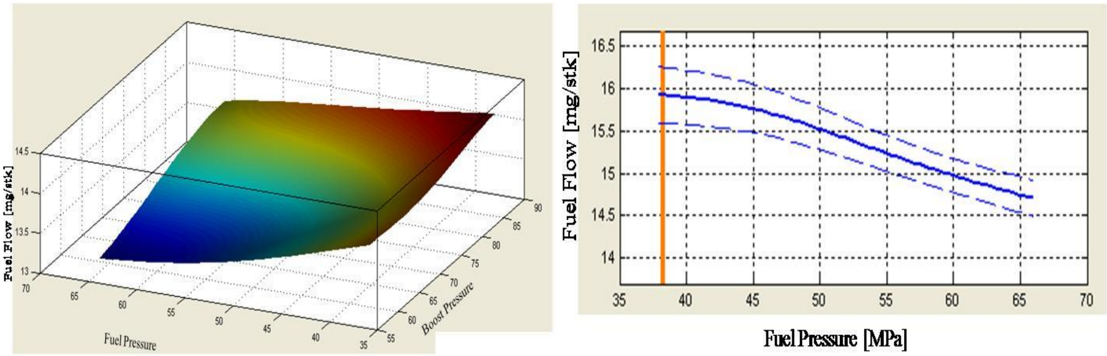
A study was carried out to evaluate the performance of the cubic polynomials by comparison with non-parametric models. In particular, this study considered Radial Basis Function (RBF) models and Gaussian Kriging.

RBF models have become particularly popular for engine response models since the Matlab MBC toolbox offers this modelling facility, and their usefulness for engine response models has been demonstrated by several studies (Morton, 2002 , Seabrook, 2007 , Sinoquet, 2009), in particular for global models. This study considered a range of kernel functions for evaluation, including Gaussian, thin plate, linear and cubic. Hybrid RBF models were also included in the study – combining a quadratic underlying model with Gaussian, thin plate, linear and cubic RBF kernels. All RBF models were fitted using the Model Browser tool in the Matlab MBC toolbox.

Kriging models were also discussed in the literature (Brahmi et al., 2009 , Kaji and Kita, 2007 , Langouet et al., 2008) as offering a useful modelling tool for engine response models, again in particular for global models. Since Kriging is an option not offered in the Matlab MBC toolbox, the commercially available JMP package was used.

Within this study comparative models were fitted for just 3 minimap points (1, 3, and 7) selected based on their equivalent residency. Evaluation of each fitted model was based on:

- Engineering trends analysis: evaluation of model adequacy in relation to expected engineering (e.g. thermodynamics) trends. While model visualisation is difficult given the high dimensionality of the space (6 input variables), both the Model Browser MBC tool and JMP provide facility for 3D visualisation (response variable plotted as a surface against 2 of the input variables, with all the other variables kept at a constant value). This is illustrated in Figure 4.6.a, based on output from the MBC. Both tools also offer the facility to plot “main effects” type plots as a 2D plot of the response variable against one of the input variables, with all the other variables constant. This is illustrated in Figure 4.6.b, based on output from MBC.
- Internal model validation: analysis of the residuals, both visual and statistical measures / diagnostics of goodness of fit including RMSE and PRESS / PRESS-RMSE for polynomial and RBF models. Given the nature of the kriging models, internal validation based on residuals does not make engineering sense.
- External model validation: analysis of prediction errors for the validation data set, both visual (residual plots) and statistical (RMSE). The analysis of residuals for the validation set is carried out automatically in MBC, whereas for the JMP Kriging models the residuals were calculated and analysed externally.



a. 3D Model visualisation b. 2D Factor effect plot

Figure 4. 6 Model Visualisation in MBC

The results from this analysis are summarised in Figures 4.7 and 4.8.

Figure 4.7 presents a comparative analysis of the performance of the different types of RBF models against the cubic polynomial, across the engine responses and engine operating points considered in the study (minimap points 1, 3 and 7). The comparison was based on PRESS RMSE as internal statistical diagnostics and validation RMSE (i.e. RMSE for the validation data set). In order to facilitate comparison across different models, the results in Figure 4.7 are expressed as error to signal ratio, i.e. [PRESS RMSE] / [Average Response], and [Validation RMSE] / [Average Response], as percentages.

The results in Figure 4.7 show that some engine responses can be modelled more accurately than others. For example, fuel flow and noise can be modelled quite accurately with PRESS RMSE and Validation RMSE below 5% of the average response. However, the emissions models are not as accurate, with PRESS RMSE and Validation RMSE above 20% of the average response.

It is also seen that modelling using RBFs can generate better models than cubic polynomials. Also, Hybrid RBF models (with a quadratic global linear model) appear to perform consistently better than the standard RBF models.

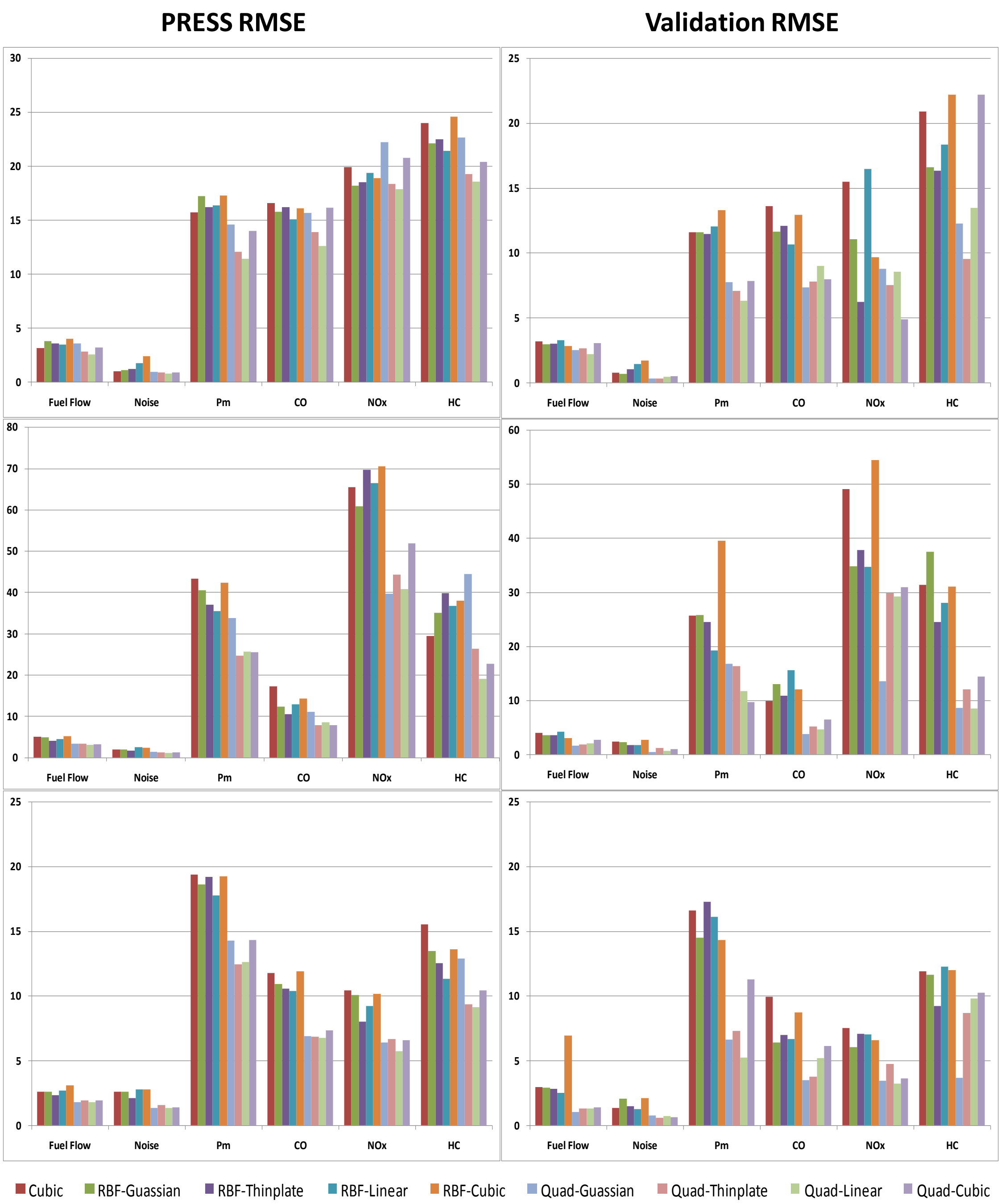
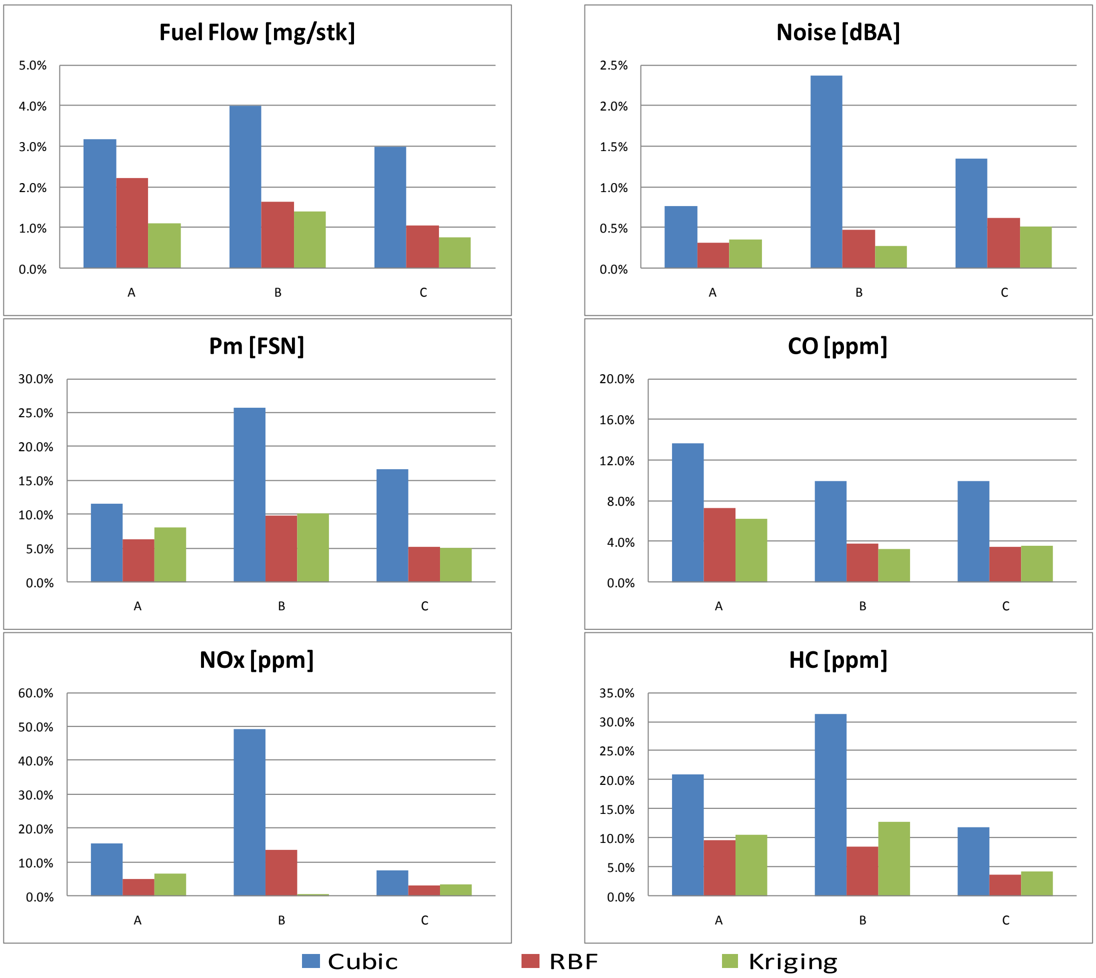


Figure 4.7 Comparative Analysis of Cubic and RBF Models, expressed as [PRESS RMSE / average response %] and [Validation RMSE / Average Response %]

Figure 4.8 summarises the comparison between cubic polynomials, RBF models (the best performing RBF model was selected) and the fitted Gaussian Kriging models. The results are presented in terms of the ratio [Validation RMSE] / [Average Response], expressed as percentage, for each engine response.

An analysis of the results in Figure 4.8 shows that the Gaussian Kriging models perform well for all engine responses and engine test points considered, and generally outperform the RBF models. This analysis also shows that both the RBF models and Gaussian Kriging models outperform the cubic polynomials, which is particularly significant for the gaseous emissions models.



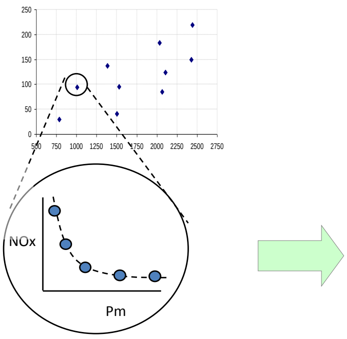
A-Minimap Point 1; B – Minimap Point 2; C – Minimap Point 3

Figure 4.8 Comparative Analysis of Cubic, RBF and Gaussian Kriging Models, expressed as [Validation RMSE / Average Response %]

4.2.4 Local and Global Optimisation

After generating the engine response models for all minimap points, the models can be used to find the optimal settings for the control variables at each minimap point. As discussed in the literature review, Diesel engine calibration optimisation is a multi-objective optimisation problem; from an external viewpoint this can be summarised as minimisation of the fuel consumption and all the emissions over the drive cycle. This suggests that optimisation must be performed both at "local" (minimap) level, and globally, i.e. over the drive cycle. The "current" optimisation process is illustrated in Figure 4.9 (Goodman, 2006b), which can be described as a two-step optimisation process, including local optimisation and global optimisation.

1. Local Optimisation



Select "optimal combinations"

2. Global Optimisation

Mode	Speed	Load	Weight (%)	NOX (g/s)	PM (g/s)
1	784	30	34.74	0.0005	0.0001
2	1009	94	3.2	0.0032	0.0002
3	1531	95	7	0.0043	0.0007
4	2065	86	3.46	0.0057	0.0008
5	2033	184	1.6	0.0372	0.0008
6	1384	137	0.93	0.0143	0.0007
7	1504	41	35.5	0.0009	0.0001
8	2443	219	0	0.0705	0.0006
9	2106	124	3.04	0.0124	0.0013
10	2428	149	0.08	0.0236	0.0013
			Cycle Totals (g)	2.6655	0.2542

Filter potential cycle results on emissions and calibration constraints

Figure 4.9 Justification of Current Two-Stage Optimisation Process

4.2.4.1. Local Optimisation

It is known that the trade-off between NOx and Particulate emissions (Pm) is the most difficult to achieve for a Diesel engine calibration (Stone, 1999). Consequently, the optimisation problem for Diesel engine calibration is normally formulated as a multi-objective optimisation problem by using Pareto based optimisation to explore the trade-off between NOx and Particulates. The Sponsoring Company uses proprietary software to perform this analysis, which implements a Normal Boundary Intercept (NBI) algorithm to solve the non-dominated search (Das and Dennis, 1998). The NBI method was used to determine the Pareto frontier identifies a specified number of points at requested intervals along the frontier.

The local optimisation task is to produce a user defined number of Pareto trade-off solutions between NOx and Particulates within the valid range. The optimisation is also subject to nonlinear local constraints on engine noise, exhaust temperature, and IMEP. The local optimisation is formulated as:

Minimise: $Obj1 = NOx\ (x)$

$Obj2 = PM\ (x)$

Subject to: $LB < x < UB$

Local constraints:

$IMEP < Limit\_IMEP$

$Exhaust\ TEMP < Limit\_TEMP$

$NOISE < Limits\_NOISE$

Equation 4.1

NOx(x) and PM(x) are the engine response models at each engine load/speed point, and LB and UB are the actuator boundaries. $IMEP$, $Exhaust\ TEMP$ and $NOISE$ are local (engine response) constraints, for which the limits are defined as $Limit\_IMEP$, $Limit\_TEMP$ and $Limits\_NOISE$.

A number of solutions, typically 5, are selected from the generated Pareto trade-off front and carried over to the global optimisation stage. The selection aims to be representative of the trade-off between the objectives, NOx and PM, thus it is based on an analysis in the decision space, and does not necessarily consider the spread of the selected solutions in the variables spaces. It is quite possible, also due to the performance of the NBI algorithm, that the solutions are not actually distinct in the variable space for some of the input variables.

Figure 4.10 shows that the current optimisation approach is carried out with two steps. Local optimisation was the first step to produce 5 Pareto trade-off solutions, which are then used as candidates for global optimisation. The second step is global optimisation, which is used to determine the combination of local optimal solutions.

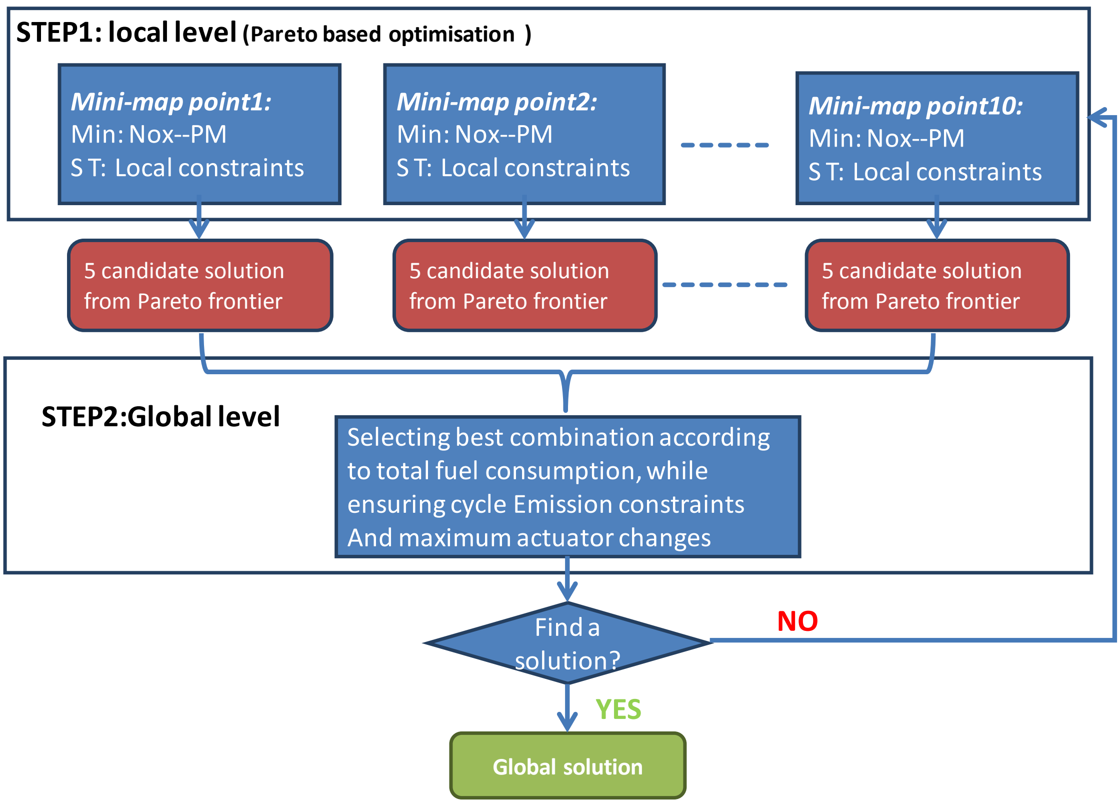


Figure 4.10 Flowchart of current calibration optimisation process

4.2.4.2. Global Optimisation

The objective of the global optimisation applied over the drive cycle, is to minimise fuel consumption (evaluated as a weighted sum of fuel consumption at each minimap point) subject to cycle emission constraints (calculated as a weighted sum of emissions at each minimap point). This is performed by exploring combinations of the ‘candidate’ solutions from the local optimisation, which were chosen from the local Pareto frontier solution sets.

An important consideration for engine calibration is to have smooth actuator maps. The implication for the global optimisation is that the actuator settings in transition from one engine load / speed (minimap) point to another should not be varied too much. Large changes in actuator settings between neighbouring minimap points could lead to significant transient behaviour and driveability issues.

To address this issue, a set of Gradient Constraints (GC) on maximum actuator changes between minimap points have been defined according to engine calibration experience. Figure 4.11 illustrates in graphical format the possible transitions between the 10 minimap points, as defined by the collaborating calibration team.

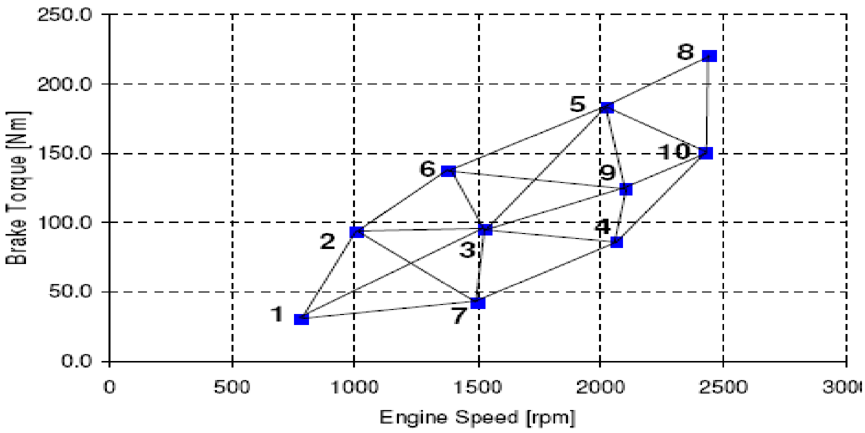


Figure 4.11 Transitions between minimap points

Figure 4.12 illustrates the way the Gradient Constraints (GC) were formulated. Considering the transition between two minimap points *i* and *j* for the same engine control, X_i and X_j, Figure 4.12 illustrates actuator settings that fulfil both the domain constraints and the actuator change constraint. The actuator change constraint can be expressed mathematically as Equation 4.2:

$$|x_i - x_j| < GC_{ij}$$

Equation 4.2

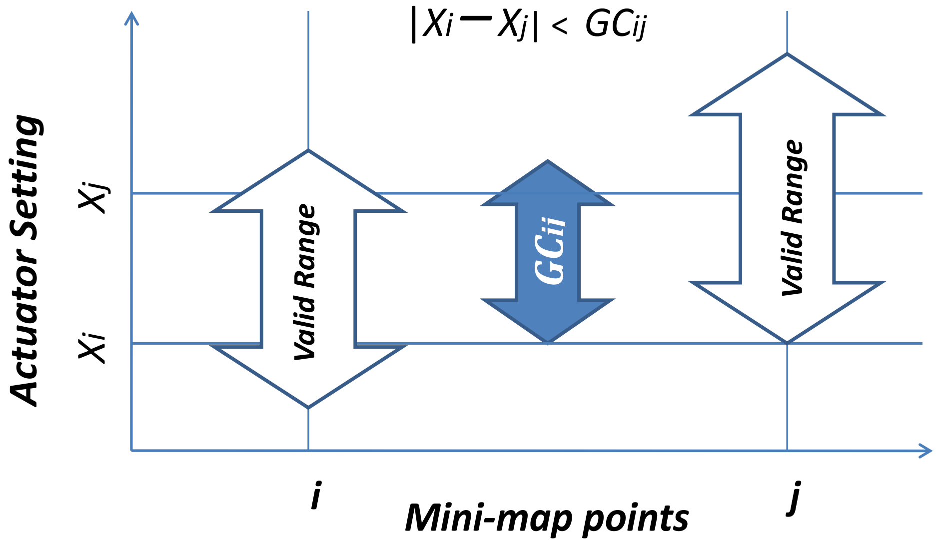


Figure 4.12 Formulation of the Actuator Change between Two minimap Points

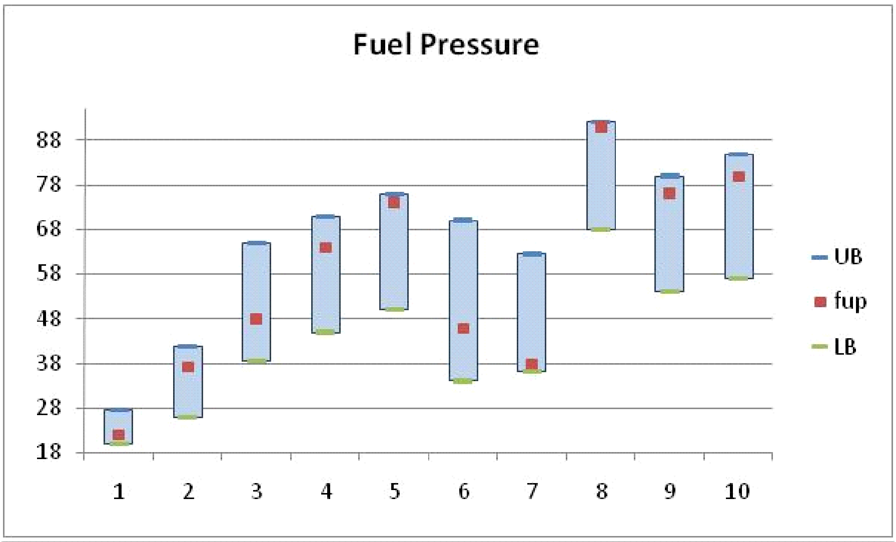
To define the GCs, an m-file was provided by sponsoring company. The m-file contains the complete set of GCs as a 21×12 matrix, which defines the limits for both decreasing and increasing each engine actuator, for the 21 possible transitions as illustrated in Figure 4.11. The reorganised GC in an Excel spreadsheet is shown in APPENDIX II.

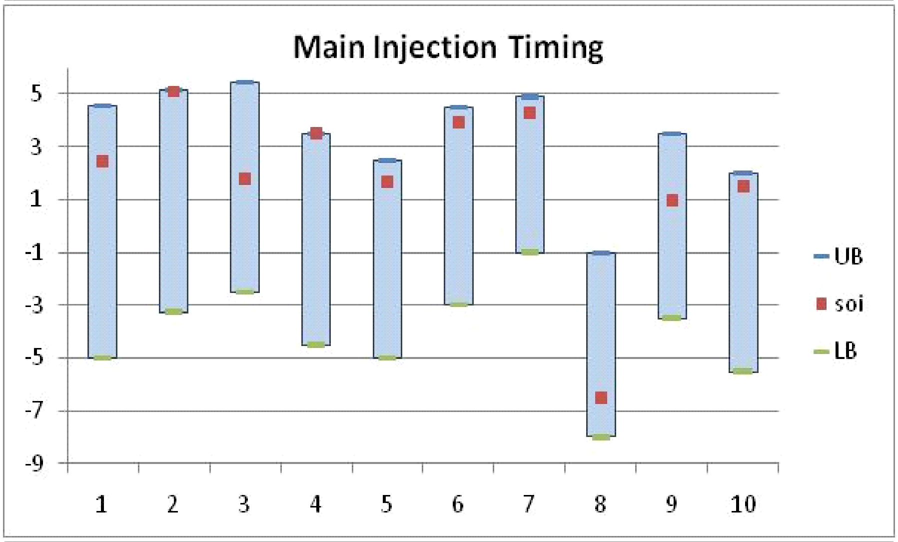
In order to carry out the global search for an optimal solution, a Matlab-based script was used (Goodman, 2006b), which performs an exhaustive evaluation of all possible combinations of candidate solutions passed on from the local optimisation stage. Given that the total number of combinations is 5^{10} (i.e. circa 10 million), this is a time consuming algorithm. The feasible solutions, which fulfil the global emissions constraints and the actuator change constraints, are sorted according to the global fuel consumption.

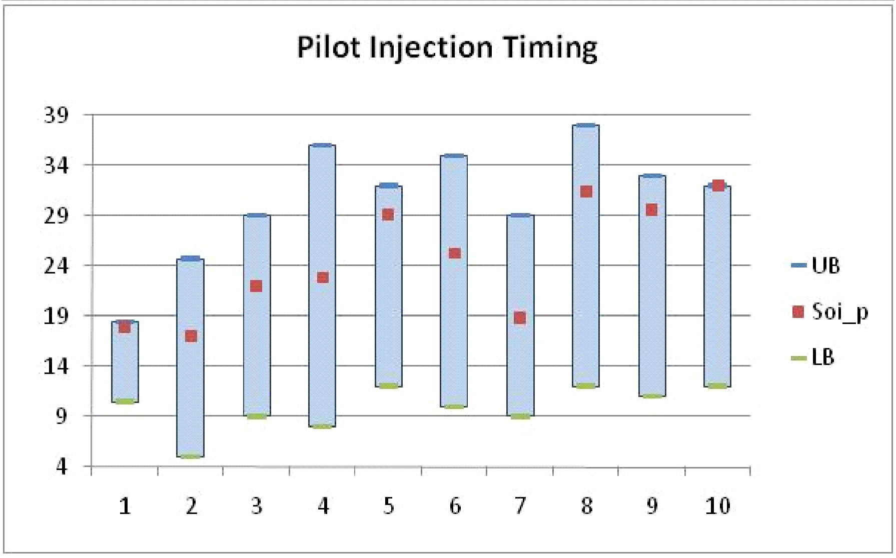
Experience with this process (Goodman, 2006b) has shown that it is quite possible that no feasible solution is found. In this case a new sampling from the local Pareto frontier is performed (i.e. 5 new points are extracted from the NOx – Pm trade-off front) and the search for the global solution is restarted. This process will be repeated until a feasible global solution is achieved.

4.2.4.3. Results

Based on the Pareto solution set, the optimal solution over the whole drive cycle has been determined throughout the second step of optimisation, while the requirements on restricting the maximum allowable engine actuator change between minimap points and on cycle emission constraints must be satisfied. In order to view the final solution in the design space, Figure 4.13 shows the actual optimal solution for each of the engine controls within the valid actuator ranges (grey bars). In Figure 4.13, it shows that each of the engine control variables has a combination of 10 optimal actuator settings for all the minimap points, which are plotted in engineering units for different engine controls.


Fuel Pressure
88
78
68
58
48
38
28
18
1
2
3
4
5
6
7
8
9
10
UB
fup
LB


Main Injection Timing
5
3
1
-1
-3
-5
-7
-9
1
2
3
4
5
6
7
8
9
10
UB
soi
LB


Pilot Injection Timing
39
34
29
24
19
14
9
4
1
2
3
4
5
6
7
8
9
10
UB
Soi_p
LB

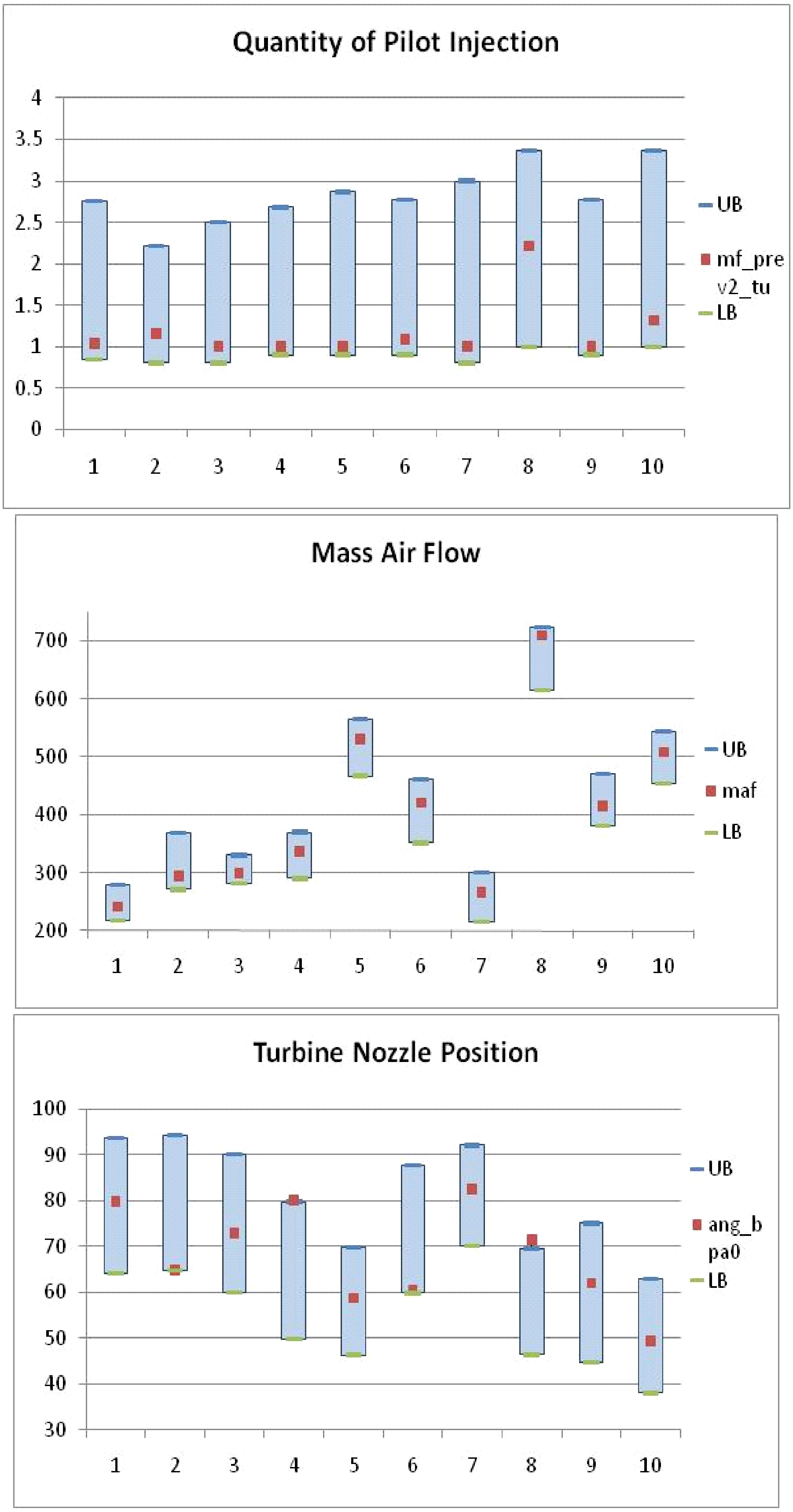


Figure 4.13 The global optimal solution plot of actuators within the valid range at different minimap points

Table 4.5 shows the estimated fuel consumption and emissions both at each minimap point and over the whole drive cycle. It shows that all the estimated emissions are within the engineering target limits for the 'hot' cycle.

Table 4. 4 The Estimated Fuel Consumption and Emissions Compare with Emission Legislation limits

Mini-map point	Fuel	NOx	CO	HC	PM
1	0.12572	0.00016	0.00557	0.00052	0.00006
2	0.01057	0.00001	0.00015	0.00001	0.00000
3	0.12020	0.00028	0.00153	0.00009	0.00010
4	0.04530	0.00014	0.00061	0.00003	0.00001
5	0.00715	0.00007	0.00002	0.00000	0.00000
6	0.02230	0.00021	0.00012	0.00001	0.00001
7	0.22771	0.00022	0.00947	0.00093	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.08438	0.00057	0.00063	0.00004	0.00002
10	0.03886	0.00032	0.00025	0.00002	0.00001
Total [kg]:	0.68218	0.00198	0.01837	0.00166	0.00021
Emission engineering targets		**0.00198**	**0.02640**	**0.00330**	**0.00044**
Fuel consumption l/100km	7.38295				
g/km		0.17998	1.66965	0.15049	0.01910

4.2.5 Validated Calibration Development Process

The output from the optimisation process is a global solution consisting of a set of actuator settings for every minimap point. The global solution is used to build up a smooth engine actuator control map by a process of interpolation, which also includes population of the relevant look-up tables.

Figure 4.14 illustrates the process used to validate this optimisation.

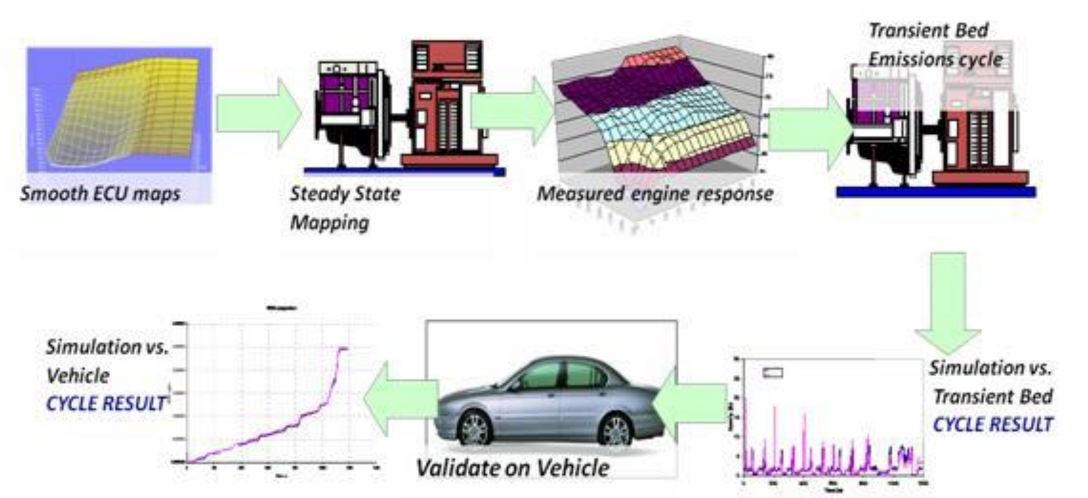


Figure 4.14 Validation Process for a Diesel Engine Calibration Approach

At first, the smooth engine actuator maps are built up by interpolation between the local calibrations for all minimap points and calibrated into an ECU. This engine control map is used to measure the engine responses on the steady state engine test bed. The collected engine emission values from the engine test bed are then compared with the corresponding estimated engine emission values from the optimisation process. To follow the steady state engine testing bed validation, the transient engine testing bed validation is also required. The transient engine test aims to estimate the total emissions over the drive cycle and compare this with the legal emission limits. Finally, the engine calibration is validated on the vehicle for the whole drive cycle.

4.3 Critique of the Current Optimisation Process

Developing a good steady state optimal calibration is a very important step in the overall calibration development process.

While the optimisation process described in the previous section is common practice in the automotive industry, as shown by the literature review (Roudenko et al., 2002 , Sheridan, 2004) and was successfully used in practice at the Sponsoring Company to generate useful steady state calibrations, it presents some shortcomings, which are outlined below:

- The optimisation is not *goal* focused: fuel economy is not actually minimised; instead, the global solution is the "best" out of the possible combinations of a subset of trade-off solutions from the local Particulates (Pm) - NOx Pareto front;
- Given that the "Global" optimisation phase relies on an exhaustive evaluation of possible combinations, this is not a computationally efficient process;
- In order to speed up the computation a small subset (typically 5) of local Pareto optimal solutions are typically selected from each minimap point as candidates for the global optimum. This not only limits the opportunities to find a true optimal global fuel consumption solution, but creates possibly frequent situations where no feasible solution can be found due to the constraints on actuator change. In this case the process is restarted with re-sampling from

the local Pareto frontiers until a global solution is found. Thus, the process can become even more expensive computationally and does not guarantee a good solution for the global optimal fuel economy objective.

Similar limitations have been found and discussed by Roudenko (2002), although based upon a different global optimisation problem formulation (i.e. a Pareto optimal trade-off at the global level between NOx and Particulates calculated as weighted sum of minimap emissions, and obtained by using the NSGA-II evolutionary non-dominated search algorithm).

A number of comparable calibration optimisation approaches have been taken in the past. Both Edwards (1997) and Haines and Dicken (2000) use very similar optimisation procedures that carry out the 'local' optimisation first and use the local optimum solution to fulfil the engine control maps. However, a smoothing procedure was required to build up the engine control map based on those local optimum solutions. One disadvantages of the current approach that is addressed earlier is the limitation of the requirement for smoothing the engine control map according to the physical limit on the actuator change over the whole range of engine operating conditions. According to these limitations, the optimisation of minimising total fuel consumption over the drive cycle cannot be simply formulated as a weighted sum of minimised fuel consumption at every engine load/speed point. Compromise is required between different engine operating points during the smoothing procedure.

Although global engine response models have been commonly used, the essential process of calibration optimisation has not been amended much. The global engine response models are used to predict engine responses at different engine operating conditions by defining equality constraints of engine, both in Atkinson's and Alonso's

(Atkinson et al., 2008) applications. Essentially the global engine response models were used same as the steady state engine local models, when the parameters of engine load and speed are fixed.

While the performance of this process could possibly be improved by better search algorithms at the global level as suggested by Roudenko (Roudenko et al., 2002), there is scope for seeking alternative formulations of the optimisation problem based on the existing data gathering and modelling process (i.e. using minimap points) that would enhance the steady state calibration optimisation performance.

Discussions with the calibration teams at the Sponsoring Company as well as other such OEMs, have pointed to further *engineering* considerations, such as:

- The two stage process itself is not ideal from a user's point of view; an "automatic" process / tool, where global optimal solutions are developed in response to the calibrator's preference expressed a priori would be preferable. While software packages such as the Matlab MBC toolbox offer strong support for the optimisation task (Styron, 2008), they do require the calibration engineer to have significant level of mathematical / statistical knowledge.
- Calibration objectives include other engineering criteria such as smoothness of responses, "engine noise" across the engine operating range, and smoothness of the actuator maps (strongly associated with driveability). The current steady state calibration process does not focus on these objectives, which are usually achieved through subsequent calibration refinement work, usually carried out on the vehicle, which hence are time consuming and expensive. If the steady state calibration could deliver "smooth" solutions, this would be likely to facilitate a shorter process and better final calibrations.

- The engine control complexity is increasing in order to address demands for better fuel economy and less pollutants; this means an increasing number of variables (e.g. multiple pilots) and constraints for the optimisation problem, with increasingly complex trade-offs, which will further decrease the efficiency of the current process. It is therefore desirable to have a formulation of the optimisation problem that can easily adapt to the increasing complexity of the Diesel engine controls, with more variables / actuators to calibrate.

4.4 Calibration Optimisation Problem Re-Analysed

From a system level perspective, the Diesel engine calibration optimisation problem has a very high dimensionality. At the system level, the global optimisation problem can be regarded as an n·k dimensional problem (where n is the number of minimap points and k is the number of control variables/actuators). For example, for the Diesel engine case study considered in this research, there are 6 actuator variables (listed in Table 4.2) and 10 minimap points (Figure 4.11), hence 60 variables.

The way in which data is collected for steady state calibration (i.e. at a set number of minimap points) defines a partition of the variables. While the control variables at each minimap point are apparently independent from the control variables at other minimap points, there is a strong coupling between these variables in terms of the actuator change constraints between the minimap points (gradient constraints), which is a form of calibrator's preference articulation for smooth actuator maps.

In terms of the decision space, it can be argued that there is a natural hierarchy of the objectives, as illustrated in Figure 4.15. The system or "global" level objectives relate to responses over the drive cycle and essentially include fuel flow and gaseous emissions (Pm, HC, CO and NOx). At “local” minimap level the objectives of interest include engine noise, exhaust temperature and IMEP.

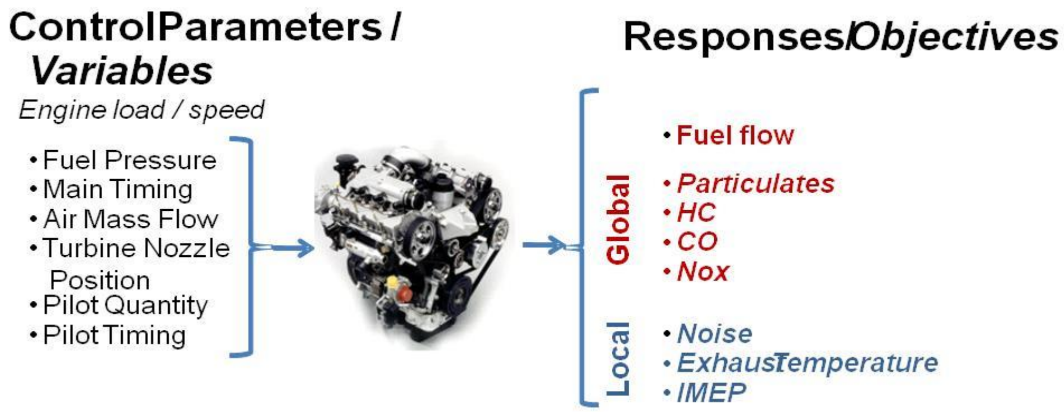


Figure 4.15 Partition of Engine Responses / Hierarchy of Optimisation Objectives

Coherent with this analysis, the calibration optimisation problem can be represented as a hierarchical multilevel structure, as shown in Figure 4.16. This figure illustrates the relationship between the system level / “global” objective and the “local” objectives, at a minimap point *i*, corresponding to a particular engine speed / load point. Within the optimisation process, the decision taken about the system variables must fulfil both the global objectives and the local objectives, which are associated with a subset of the global variables (actuator settings at a minimap point). This implies that within the optimisation process there must be strong co-ordination between the global and the local objectives.

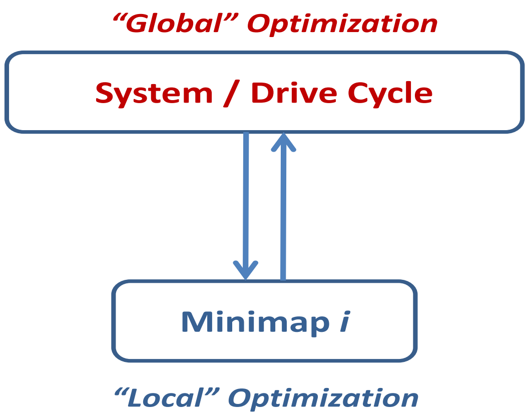


Figure 4.16 Multi-level Structure of Calibration Optimisation Problem

The analysis in Figure 4.16 does not reflect the coupling between variables discussed earlier, associated with the requirement for smooth actuator maps. Figure 4.17 illustrates a revised analysis, showing the coupling between variables associate with minimap points at the "local" level, in terms of the gradient changes constraints.

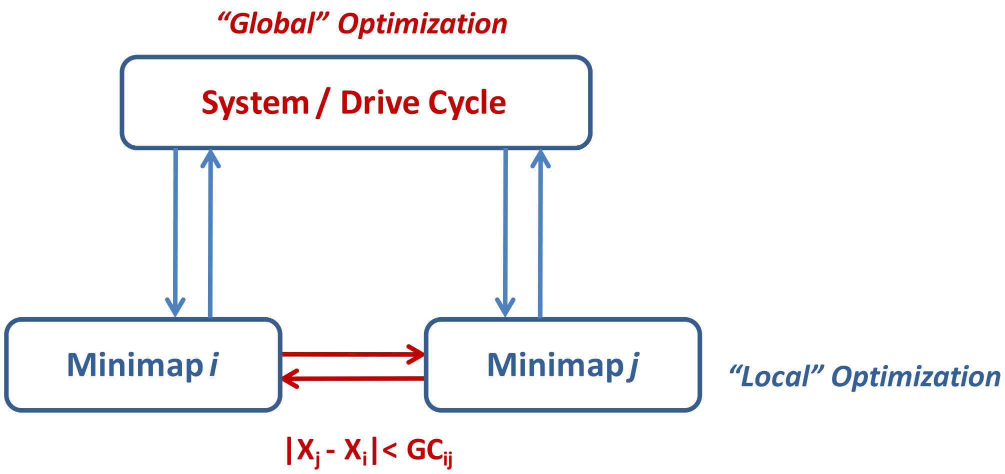


Figure 4.17 Multi-level Structure of Calibration Optimisation Problem, including Coupling Between Variables

The Literature Review (Chapter 2) discussed Multidisciplinary Design Optimisation (MDO) frameworks, which have been introduced as a practical approach for dealing with modern engineering systems such as aircraft and automotive vehicles, which are increasingly complex, with high dimensionality (more than 100 input variables) and strong coupling interactions.

The analysis presented in this section shows that the calibration optimisation problem could be regarded as being in the Multi-disciplinary Design Optimisation (MDO) framework. This makes a strong case for attempting to formulate the calibration optimisation problem within an MDO framework, which has been demonstrated as having strong benefits in terms of simplifying and reducing the cost of the analysis in other application areas in aerospace and automotive engineering.

According to the literature (Braun et al., 1996), All At Once (AAO) is an immediate opportunity for MDO problem. In Chapter 2, an AAO optimisation architecture has been illustrated (Figure 2.5) and discussed. As a highly centralised framework, AAO framework offers strong capability to handle both global level targets and local level/discipline targets, where in this case global targets refer to total fuel consumption, cycle emissions and actuator map smoothness over the whole cycle, local level/discipline targets refer to local constraints. Figure 4.18 illustrates the possible AAO structure formulation for the Diesel engine calibration problem, according to the suggested AAO architecture (Figure 2.5) from the literature (Braun et al., 1996).

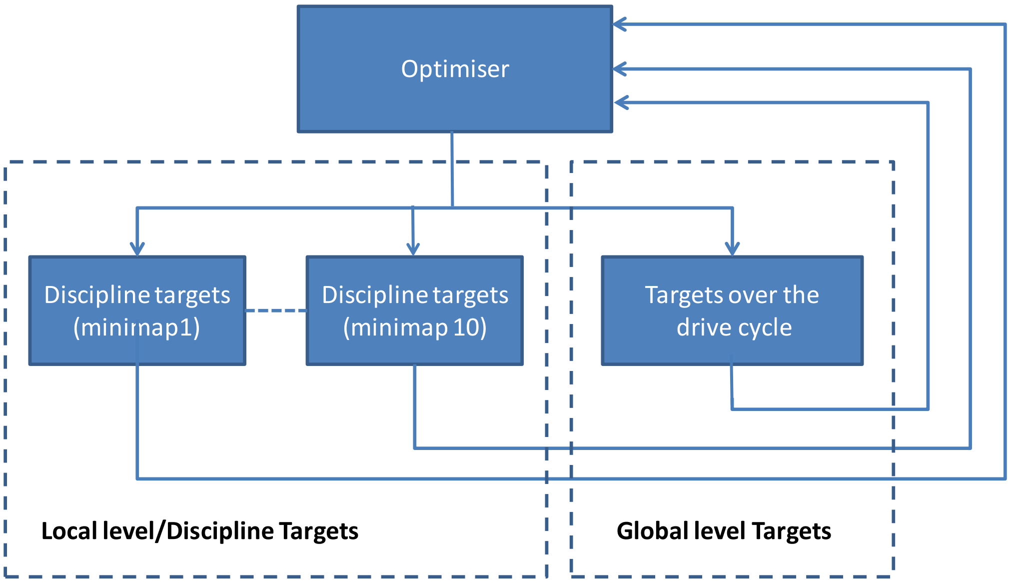


Figure 4.18 All At Once Optimisation Architecture for Diesel Engine Calibration Problem

Of the other MDO framework in literature (Braun et al., 1996 , Kroo et al., 1994 , Kroo and Manning, 2000). The Collaborative Optimisation framework as a multi-level MDO structure is very popular for dealing with situations where there is a strong coupling between variables, as discussed in Chapter 2 (Section 2.4.2.1). Therefore, MDO/CO appears to have more potential to handle the Diesel engine calibration optimisation problem based on the analysis of the Multi-level structure of this problem. Due to the analysis of the Diesel engine calibration system as a multi-level problem, it may be concluded that the steady state Diesel engine calibration problem can easily fit into MDO/CO framework according to the illustration in Chapter 2 (Figure 2.8). Figure 4.19 demonstrates the possible MDO/CO framework formulation for the steady state Diesel engine calibration problem.

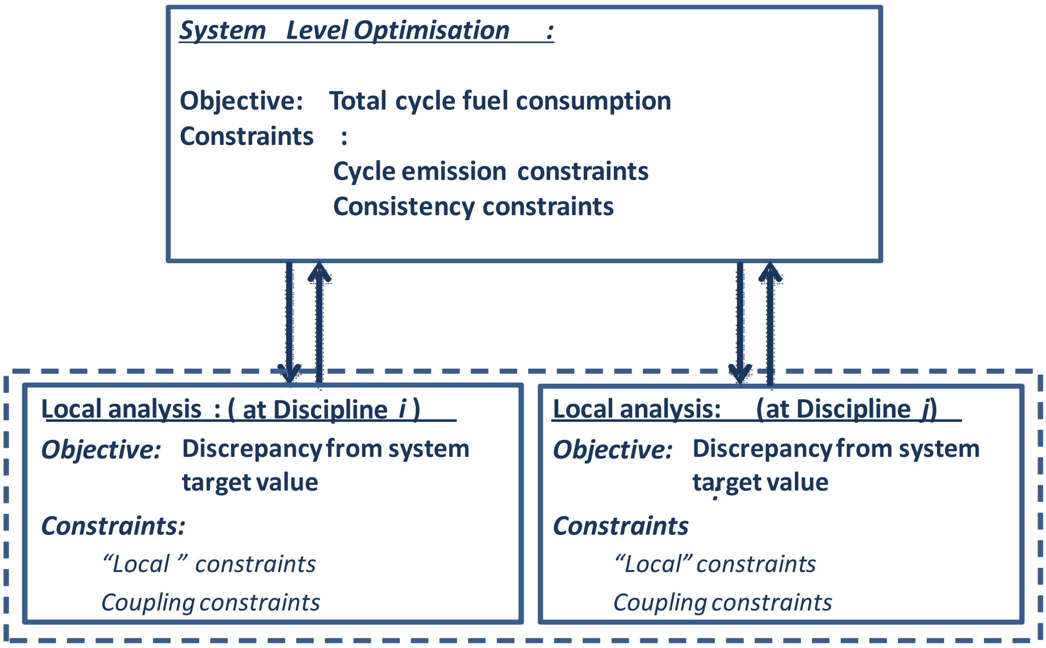


Figure 4.19 MDO/CO Framework Architecture

Due to the main interest of engine calibration process is to reduce the fuel consumption and emissions over drive cycle, they are naturally formulated as global targets locate at the top level. In the MDO/CO structure suggested, the discipline optimisation is to minimise the discrepancy between the subsystem/local variables and system/global target values while treating the local targets as constraints. Another one of the benefit of the MDO/CO framework is the decomposition of the coupling between disciplines, which relate to the actuator gradient change constraints. Therefore, a possible choice is to decompose the Diesel engine calibration problem into a number of disciplines that are defined by minimap points.

4.5 Summary

Based on the analysis presented in the previous section, the All At Once (AAO) and Collaborative Optimisation (CO) MDO framework appear to be well suited for the Diesel engine calibration optimisation problem. The following Chapters describe the development, implementation and testing of these frameworks for the Diesel engine case study considered in this research.

Chapter 5 “All At Once” Multi-disciplinary Design Optimisation Approach to Diesel Calibration Optimisation

5.1 Introduction:

This chapter aims to describe the development and implementation of an AAO framework for the Diesel engine calibration optimisation problem based on steady state models.

“All At Once” (AAO) is the most basic MDO method and is a highly centralized approach which attempts to analyse and solve simultaneously both the global and local problem. Figure 5.1 illustrates a simplified structure for an AAO framework; the “Global Optimiser” focuses on optimising the main objective (or objectives / targets) for the engineering system, while the “System Analyser” evaluates the global (system level) requirements and the “local” (discipline level) targets. Within the AAO framework the global analysis and the local analysis are performed at same time, dealing with all variables simultaneously, thus ensuring that the optimisation process ends up with both the global and local requirements (targets and constraints) being satisfied.

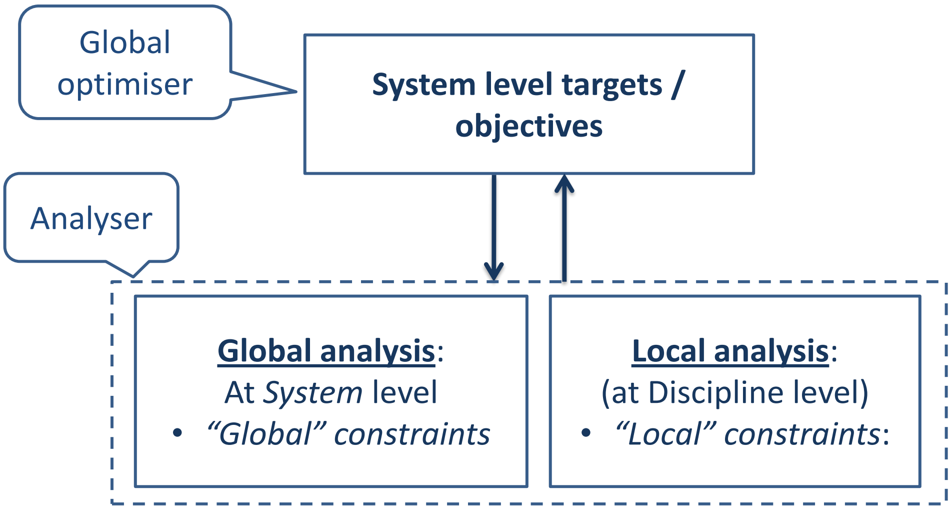


Figure 5.1 Illustration of AAO Framework

5.2 Diesel Engine Calibration Problem Analysis and Formulation as MDO / AAO

According to the discussion in Chapter 4 (Section 4.5), the steady state Diesel mapping and calibration optimisation can be presented as a 2-level hierarchical structure, as illustrated in Figure 4.17. It has been argued that this problem can be represented within an MDO framework in which the local calibration problem (optimal actuator settings at each minimap point) can be regarded as "disciplinary" level, while the over the drive cycle outcomes can be regarded as the system or global objective and / or targets.

The calibration analysis at each minimap level, discussed in Section 4.2.2, highlighted the need to ensure combustion stability, engine noise performance and after-treatment efficiency. From an optimisation point of view, the calibration preferences for these attributes have been formulated in terms of targets, which in turn can be mathematically formulated as constraints, as shown in Equation 4.1.

From a global / system level point of view, it can be argued that cycle fuel consumption (i.e. the total fuel flow needed to achieve the required torque at all engine operating conditions) is ultimately the most important attribute, while gaseous emissions output over the drive cycle can be constrained to an engineering target (limit) level (discussed in Section 4.2).

The smoothness of the actuator maps has also been discussed as an important calibration criteria, directly associated with refinement and drive-ability, which is a system / global level attribute / objective. Within the "current" calibration process, this calibration target has been formulated in terms of a maximum actuator change allowed for transitions between minimap points, as discussed in Section 4.2.4.2.

Table 5.1 summarises this analysis of engineering objectives and targets associated with the calibration optimisation problem.

Table 5. 1 AAO optimisation – engineering analysis of objectives and targets

	Discipline / Local level	**System / Global level**
Objectives		• *Minimise Fuel consumption over the drive cycle*
Targets	• Combustion stability (IMEP); • Engine Noise (Apollo); • Exhaust Temperature.	• Engineering limits for gaseous emissions (NOx, PM, HC, CO); • Drive-ability – Smoothness of actuator maps – gradient / actuator change constraint.

Figure 5.2 illustrates the structure of the Diesel steady state calibration optimisation problem analysed within an MDO / AAO framework.

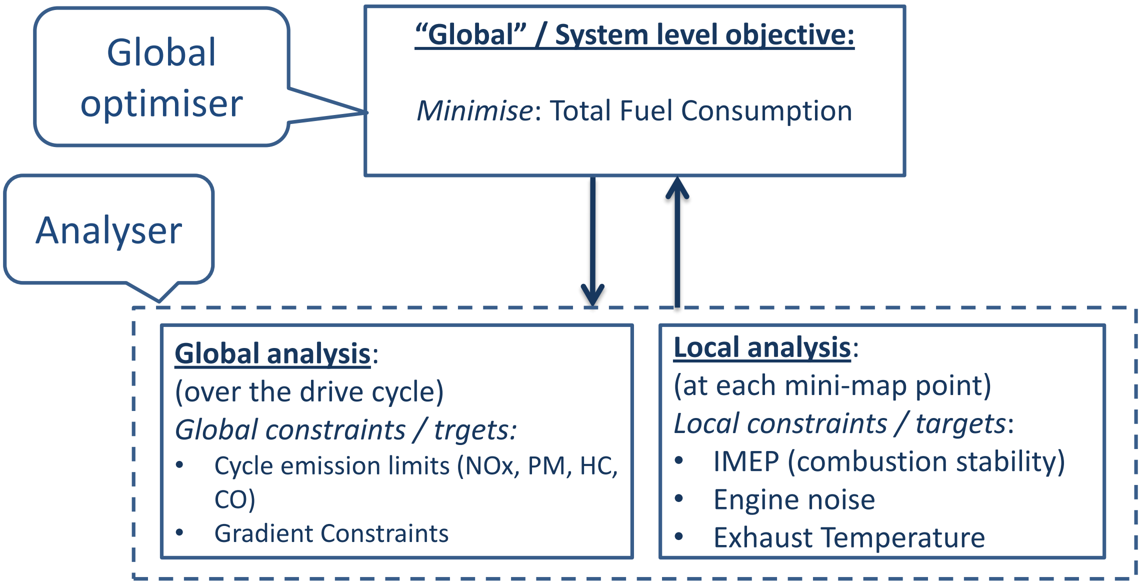


Figure 5.2 AAO Formulation for Steady State Diesel Engine Calibration Problem

Within the AAO framework, the "global optimiser" focuses on minimising total cycle fuel consumption and at the same time the "analyser" simultaneously performs all the evaluations of the current performance against the targets imposed by the global and local analysis.

The mathematical formulation of the AAO optimisation problem is shown in Equation 5.1.

Minimise: $J(X) = \sum_{i=1}^{n} w_i * F_i(X) \qquad i \in (1, \dots n)$

Subject to:

Global constraints:

$$G(X_i) = \sum_{i=1}^{n} w_i * Emission_i < Emission\_Limit$$

$$GC\left(X_i, X_j\right) = \left(X_i - X_j\right) < GC_{limit_{ij}} \qquad i \neq j, j \in (1, \dots n)$$

Local constraints:

$$H(X_i)_i = Local\ \_constraint_i < Local\_constraint\_limit_i$$

$$LB_i < X_i < UB_i$$

Equation 5.1

Equation 5.1 uses the following notation:

- X_i is a k-dimensional vector of calibration variables at minimap point i; where k is the number of actuators, 6 in this case as listed in Table 4.2, where $i \in (1, \dots n)$, where n is the number of minimap points, 10 in this case;
- $F_i(X_i)$ is the Fuel consumption at node i;
- $J(X_i)$ is the Objective function, expressed as the weighted sum of fuel flow in L/100Km;
- w_i is the Residency time associated with minimap point i;
- $G(X_i)$ is the Global constraint, which is a vector of cycle emissions, evaluated as the weighted sum of emission output at each minimap point i;
- $Emission_i$ is a vector of gaseous emissions evaluated at minimap point i, i.e.

$$Emission_i = \begin{bmatrix} NOx[X_i] \\ Pm[X_i] \\ HC[X_i] \\ CO[X_i] \end{bmatrix}$$

- $Emision\_Limit$ is the vector of engineering cycle limits (or target) for gaseous emissions, i.e.

$$Emission\_limit = \begin{bmatrix} NOx\_limit \\ Pm\_limit \\ HC\_limit \\ CO\_limit \end{bmatrix}$$

- $GC(X_i, X_j)$is an $n \times k$ matrix of the actuator changes between neighbouring engine operating points i and j;
- $GC_{limit_{ij}}$ is the $n \times k$ matrix of maximum allowable actuator change for each actuator and transition between minimap points;
- $H(X_i)_i$ is the Vector of evaluated 'local constraints' for node i,
- $Local\_constraint$ is the vector of local responses evaluated at minimap point i, i.e.

$$Local\_constraint_i = \begin{bmatrix} IMEP[X_i] \\ Apollo[X_i] \\ Ex\_Temp[X_i] \end{bmatrix}$$

- $Local\_constraint\_limit$ is the vector of targets for local responses at minimap point i, i.e.

$$Local\_constraint\_limit_i = \begin{bmatrix} IMEP\_limit \\ Apollo\_limit \\ Ex\_Temp\_limit \end{bmatrix}$$

- LB_i, UB_i are Domain constraints that is vector of lower bounds (Alberer) and upper bounds (UB) for the design variables X_i at minimap point i;

Consequently, this engine calibration optimisation problem has 60 design variables and 316 constraints, made up of:

- 4 global emissions constraints;
- 252 gradient constraints;
- 60 local constraints.

5.3 Algorithm Development and Implementation in Matlab

The analysis above has revealed the complexity of the optimisation problem. To solve such a large dimensional design space and heavily constrained optimisation problem, an effective algorithm needs to be employed. Based on the discussion of the literature review in Chapter 3, the GA is more capable of dealing with large scale optimisation problems with a complex design space (more curvature and constraints) than the gradient based searching methods. GA is also well established with many applications (Alonso et al., 2007 , Brooks and Lumsden, 2005 , Rask and Sellnau, 2004 , Roudenko et al., 2002) rather than other Evolutionary Algorithms, such as

PSO. Thus, an evolutionary Genetic Algorithm approach has been selected for implementation.

Matlab was selected as the development and implementation environment in order to ensure compatibility with the other calibration support tools used by the Sponsoring Company. An existing MATLAB based GA tool was initially used for the AAO implementation.

5.3.1. Development of Data Structures for AAO Implementation

In order to develop and run an efficient optimisation algorithm an appropriate data structure to manage the AAO optimisation problem must be created (including design variables, engine response models and constraints).

The Matalb programming environment offers a variety of data structures, such as ‘double’ for Floating-point number, cell array, structure and function handle and so on. The structure of the data is designed to carry information according to minimap points. Table 5.2 summarises the m-files created to manage the data structures for engine calibration optimisation.

Table 5.2 Engine Calibration Optimisation Inputs

File Name				
Data / Description	Actuator Change Gradient Constraints	Inputs variable limits	Cycle Emission Limits	Local engine output constraints
Data Structure	Cell array	Cell array	Structure	Cell array
File Name	'mm_n'	'mm_t'	'Models'	'res_time'
Data / Description	Engine speed for each minimap points	Engine torque for each minimap points	Engine response models	Residency time for each minimap points
Data Structure	Double	Double	Cell array	Double

Table 5.2 shows that a number of data is in "cell array" format. A cell array can store the data in the format of a structure or matrix in each element, which also allows organising the data sets from different minimap points within one cell array. Using structure files allows the storing arrays of varying classes and sizes, and names to denote contents and simple methods of passing function arguments or models. A number of data files are designed with a special structure, which is explained as follows:

- As described in Section 4.2.4.2, the original 'GC' is a matrix that involves 21 transitions. For the AAO implementation the 'GC' matrix was reorganised into a cell file. Figure 5.3 shows the structure of the 'GC' cell file that involves actuator change limits for all the existing transitions. This is a clearer structure to represent the existing transitions between minimap points and is easier to use compared with the original 'GC' data format.

Fup		soi_main1		soi_prev2		mf_prev2		maf		bpapwm	
LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
-2	15.75	-5.37	5.37	-8.25	8.25	-0.2	0.85	-160	160	-19.5	19.5

		Transition minimap point									
		1	2	3	4	5	6	7	8	9	10
Transition minimap point	1		<1x12 double>	<1x12 double>				<1x12 double>			
	2	<1x12 double>		<1x12 double>			<1x12 double>	<1x12 double>			
	3	<1x12 double>	<1x12 double>		<1x12 double>	<1x12 double>	<1x12 double>	<1x12 double>		<1x12 double>	
	4			<1x12 double>				<1x12 double>		<1x12 double>	<1x12 double>
	5			<1x12 double>			<1x12 double>		<1x12 double>	<1x12 double>	<1x12 double>
	6		<1x12 double>	<1x12 double>		<1x12 double>				<1x12 double>	
	7	<1x12 double>	<1x12 double>	<1x12 double>	<1x12 double>						
	8					<1x12 double>					<1x12 double>
	9			<1x12 double>	<1x12 double>	<1x12 double>	<1x12 double>				<1x12 double>
	10				<1x12 double>	<1x12 double>			<1x12 double>	<1x12 double>	

Figure 5.3 Illustration of Gradient Constraints Data Structure

- 'EM_Lims' is a structure file of estimated emission limits with emission names.
- Apart from 'GC' and 'EM_Lims', the data files are in either a matrix or a cell file structure, which are organized in the structure according to the mini-map. For example, the engine response models were built up in the MBC software algorithm and exported into the workspace in MATLAB. The engine response models are stored in the 'Models' cell file corresponding to different minimap points. Each of the elements in this cell file defines a set of engine response models.

Figure 5.4 explains the data requirements for AAO implementation by applying the standard GA algorithm in MATLAB. The main implementation involves three sections viz. data loading, data sorting and optimisation, as shown in APPENDIX III.

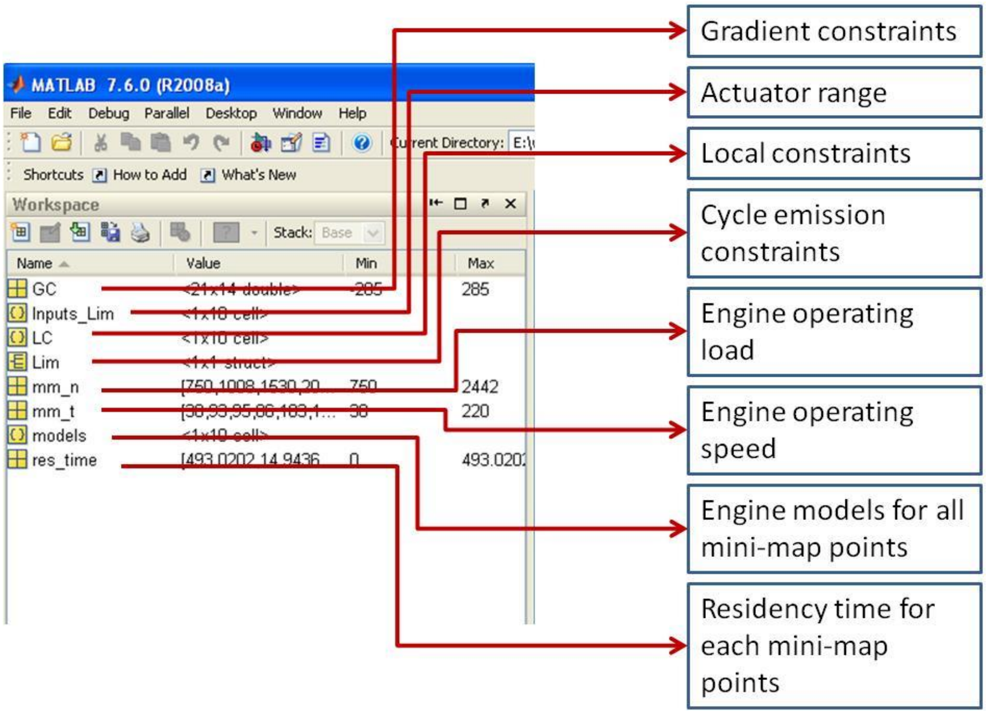


Figure 5.4 Explanation of Data Requirement for AAO implementation with GA algorithm

5.3.2. MATLAB Implementation

In order to run the optimisation, the objective and constraints functions need to be defined. As shown in Equation 5.1, the objective function is formulated as the total fuel consumption, calculated as the weighted sum of fuel consumption from each minimap point. This was implemented in the m-file called ‘objective’, which is shown in APPENDIX III. In Figure 5.2, the constraints function was described as the formulation that includes global constraints and local constraints. The constraints function was also developed into a m-file called ‘con_sys’, which is shown in APPENDIX IV.

Figure 5.5 illustrates the Matab implementation.

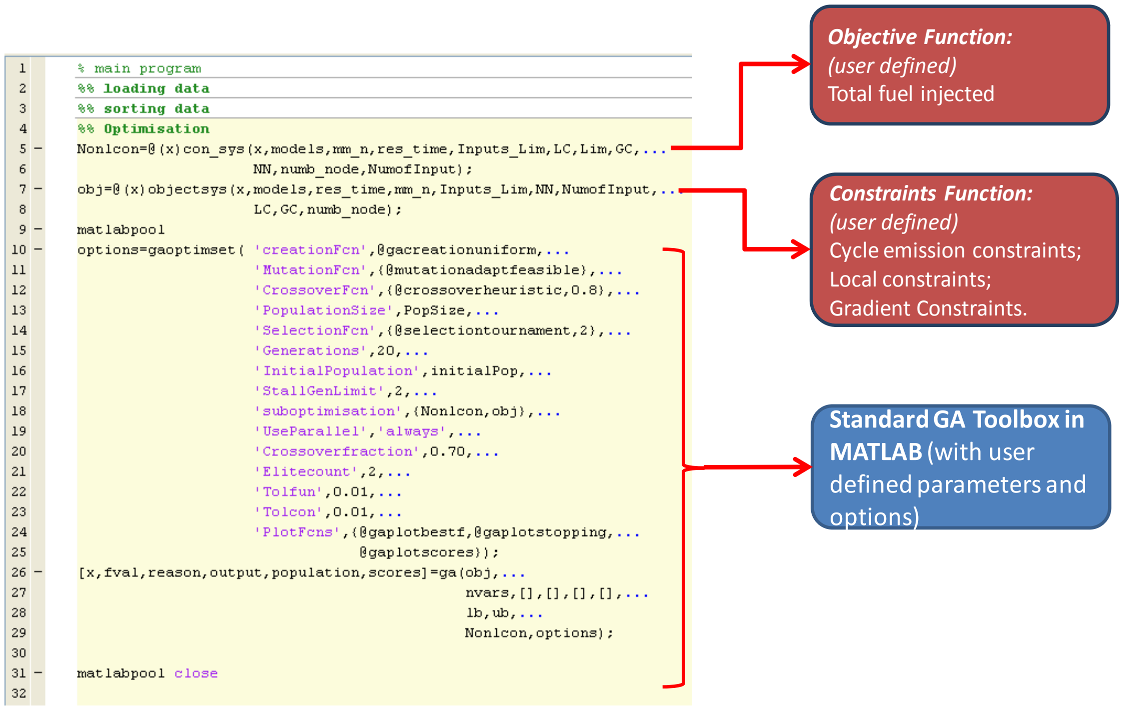


Figure 5.5 Demonstration of AAO Implementation by Applying GA Standard Algorithm

The GA programme requires that the user defines the GA options for different optimisation problems. Table 5.3 summarises the GA options set for the AAO optimisation.

Table 5.3 Algorithm option setting for 3 minimap points test

Option name	Setting	Description and explanation
Selection Fcn	selectiontournament,2	Randomly select a number of solution from the current population, and the best individual is selected and saved as parents solution for next generation
PopulationSize	Twice of input variable size	To define the number of individuals in the population. It must be chosen with care. If it is too low, it cannot explore the design space enough, otherwise the selection perform poorly with higher possibility to choose similar solutions. Normal suggestion is 1.5 or 2 times of the number of variables
Generations	200	To define the total generation for the optimisation process, it very much depends on the convergence speed.
Stall Gen Limit	4	To define the stopping criteria by the number of iterations without improvement in the objective values in the population, for example 4 of consecutive generations is used.
Elite count	2% of population size	Use a small percentage of the population to maintain the best achievement while maximally keep the diversity
Crossover fraction	0.6	To keep the searching route within the reasonable distance or local area
TolCon	0.01	Reasonable tolerance is chosen according to the emissions engineering unit (ppm)
TolFun	0.01	Reasonable tolerance is chosen according to the objective engineering unit (mg)

The population size is defined as twice of input variables size in order to explore the design space well. The “tournament” selection function was used to randomly select a number of solutions from the current population that is defined as 2 in this application, and the best individual is selected and saved as parents solution for next generation until enough parents solution are selected. For every generation 2% of the top population solutions were kept for the next generation and 60% of the population was generated by the crossover operator. The remaining solutions were generated by the mutation operator.

5.4 Preliminary Results and Analysis

To assist with the development and implementation of the algorithms and to obtain an early evaluation of the potential of the developed AAO optimisation framework, the methodology was applied initially to a reduced set of minimaps consisting of only three points. These points were selected from the 10 minimap points by choosing the points with highest residency times, shown in Table 5.4. The equivalent residency time was re-calculated using the same approach as for the original 10 points, i.e. the weight of each point in the simulated NEDC "cloud" (Figure 4.3) was added to the closest of the selected minimap point.

Table 5.4 Three Selected minimap for Pre-test

Number of points	Engine load (Nm)	Engine speed (rpm)	Residency time (s)
Mini-map Point 1	30Nm	750rpm	493.97 s
Mini-map Point 2	95Nm	1530rpm	231.03 s
Mini-map Point 3	42Nm	1505rpm	370.98 s

Testing the optimisation procedure on a reduced data set was also a strategy adopted by the Sponsoring Company, therefore benchmark results are available to compare the AAO results against a typical output from the "current" optimisation process described in Chapter 4.

5.4.1 AAO Results for Three Minimap Points

The AAO optimisation was run with the settings described in Table 5.3. Figure 5.6 illustrates the convergence plot for the GA optimisation process, stopping criteria and individual objective values of the whole population. From the convergence plot, it can be seen that there is a rapid reduction during the first few generations and then the convergence curve is flatten. The stopping criteria show the objective value (fuel consumption) has not been improved more than the tolerance after the 65th generation.

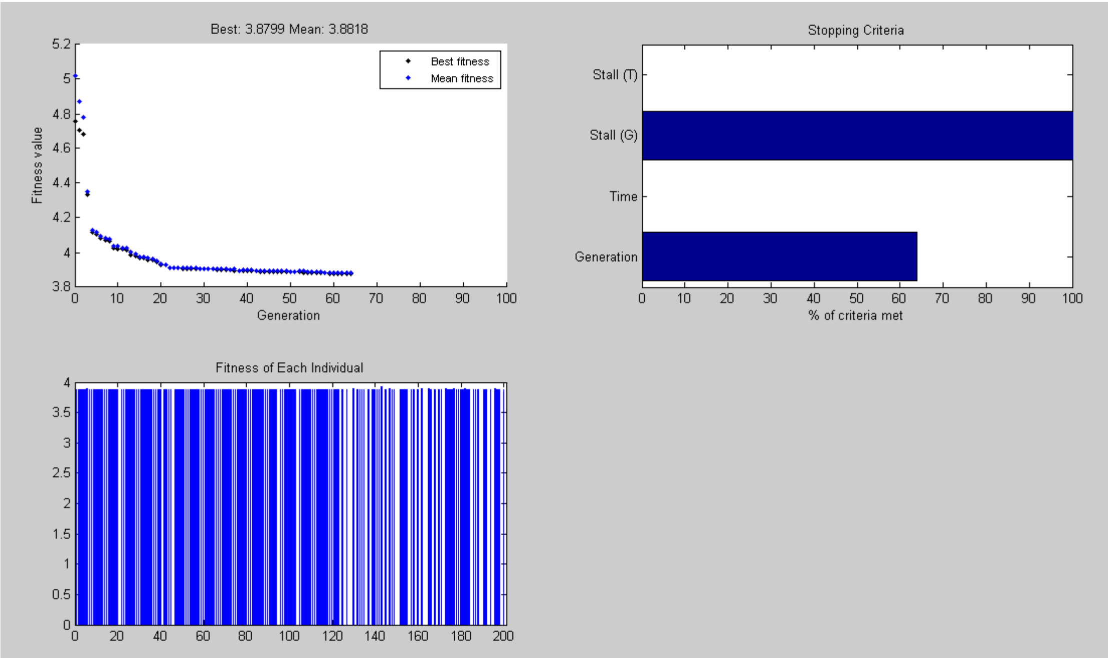


Figure 5.6 Illustration of GA Convergence with Stopping Criteria and Individual Fitness Values

5.4.2 Study of Effect of Population Size on Algorithm Performance

From the literature review of optimisation (De Jong and Spears, 1991), the minimum population size for a GA algorithm was suggested to be set as 1.5 or 2 times the number of control variables. This also depends on the complexity of the design space. Given the large dimensionality of the calibration optimisation problem, as reflected by relatively high computation expense observed for the AAO, a question arises whether there is a trade-off between GA population size and the performance of the algorithm in terms of the quality of the solutions and the computation time.

A study was conducted based on a set of tests for different population sizes. Four population sizes were chosen as 50, 100, 200 and 500, and for each population size setting, three repeat runs of the programme were carried out. Table 5.5 summarises the results for fuel consumption and computation time.

Table 5. 5 AAO testing results for three nodes

Popsize	Test Num	Fuel consumption (L/100Km)	Computation (hrs)
50	1	3.98	1.474
	2	4.10	2.178
	3	3.90	1.997
	Average	3.99	1.8833
100	1	3.93	5.106
	2	4.04	3.399
	3	3.93	4.541
	Average	3.97	4.3483
200	1	3.88	5.561
	2	4.10	7.795
	3	3.95	8.236
	Average	3.98	7.1973
500	1	3.46	19.478
	2	3.97	15.123
	3	3.98	21.700
	Average	3.80	18.7672

In order to make the results more easily understood by calibration engineers, the fuel consumption values have been transferred from units of ‘mg’ to ‘L/100km’ by Equation 5.2:

$$\boldsymbol{F(L\,/\,100Km) = \ (f_{tot}\,mg * 100Km)/(840\ g\,/\,L * 1000 * 11Km)}$$

Equation 5.2

Table 5.5 shows that the increase in population size costs more expensive computation and does not necessarily result in a significant improvement.

5.4.3. AAO Results Analysis

The AAO optimisation results were compared with the benchmark results obtained from the use of the conventional ("current") process described in Chapter 4, for the same reduced data set (3 minimap points).

In terms of the main objective of the optimisation, Figure 5.7 shows that AAO optimisation delivered a significant (12.6%) improvement in fuel consumption compared with the current two-step optimisation process.

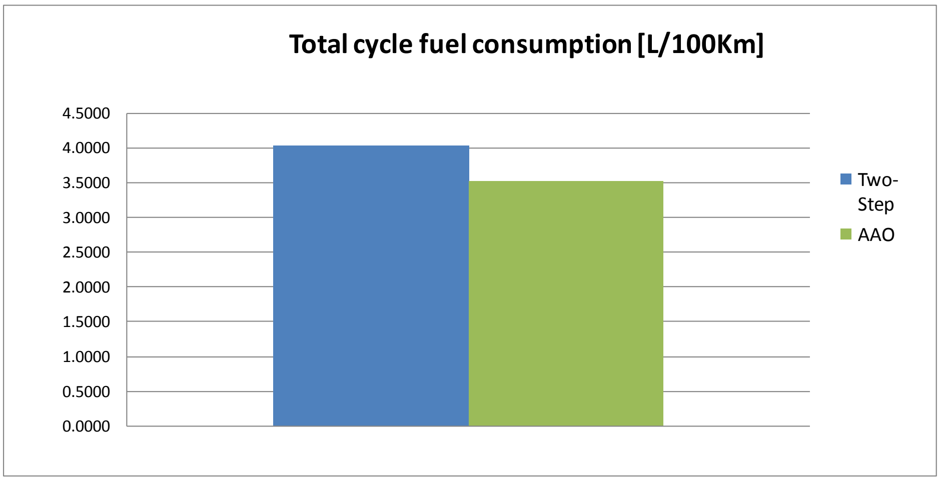
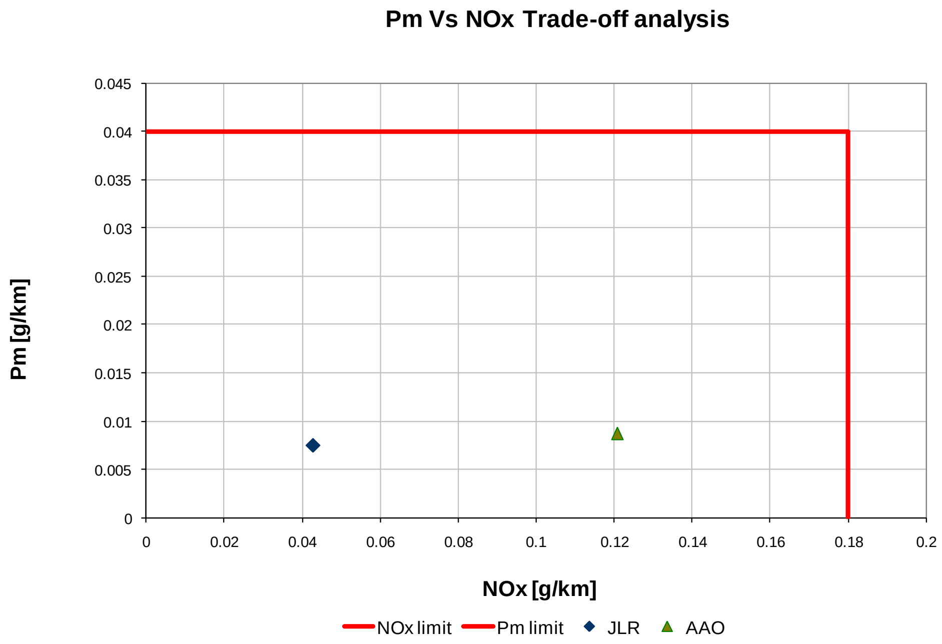


Figure 5.7 Results Analysis (Fuel Consumption): Two-Step vs. AAO

Figure 5.8 presents comparative results from both the two-step approach and the AAO approach in terms of cycle emissions. For convenience these results are presented in terms of NOx-PM and HC- CO trade-offs, with the engineering targets for total cycle emissions also indicated on the charts. It shows that both of the optimisation approaches produced valid solutions within the cycle emissions constraints. However this is only based on the three minimap points simulation test.



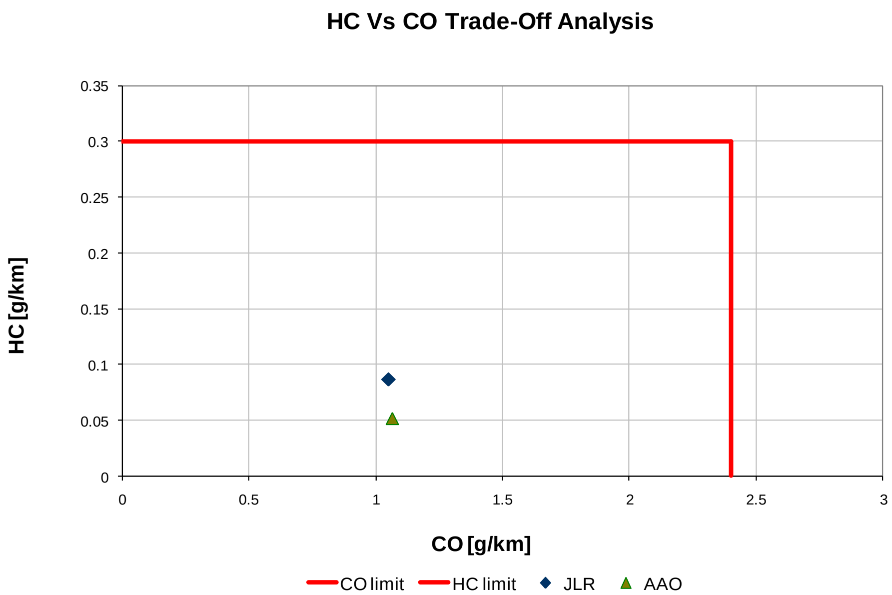
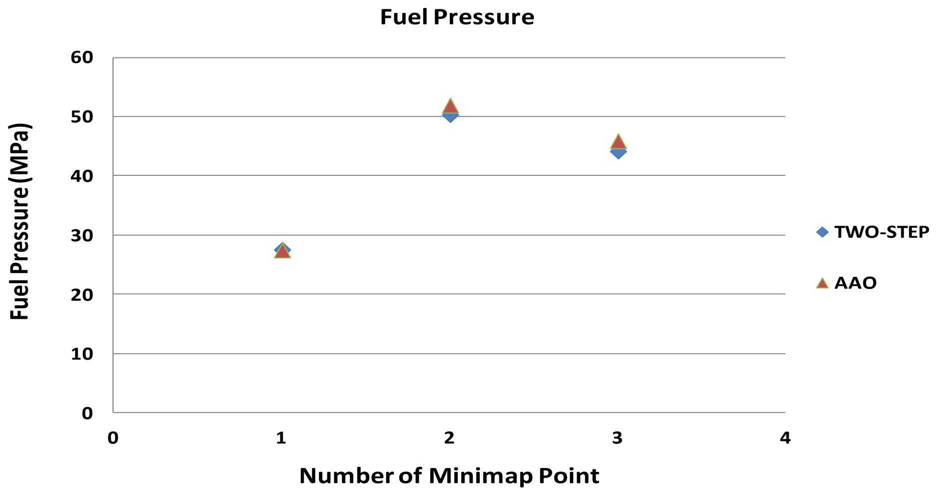
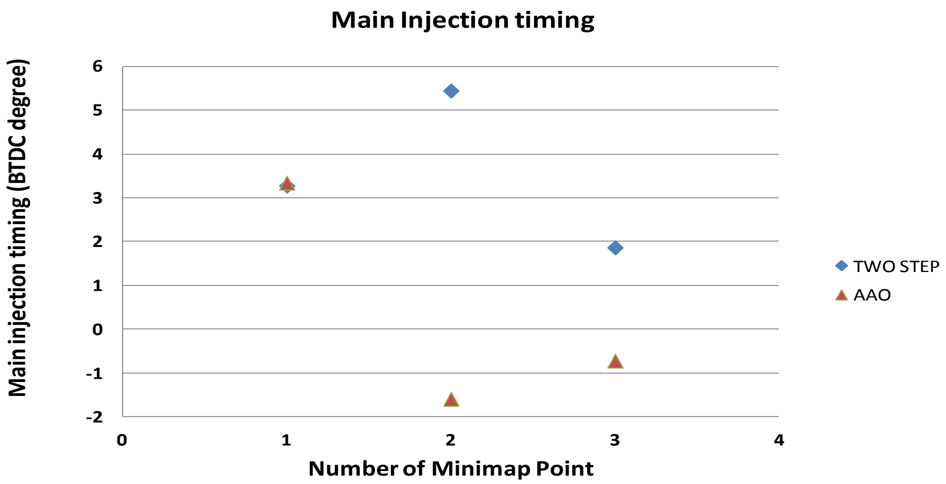
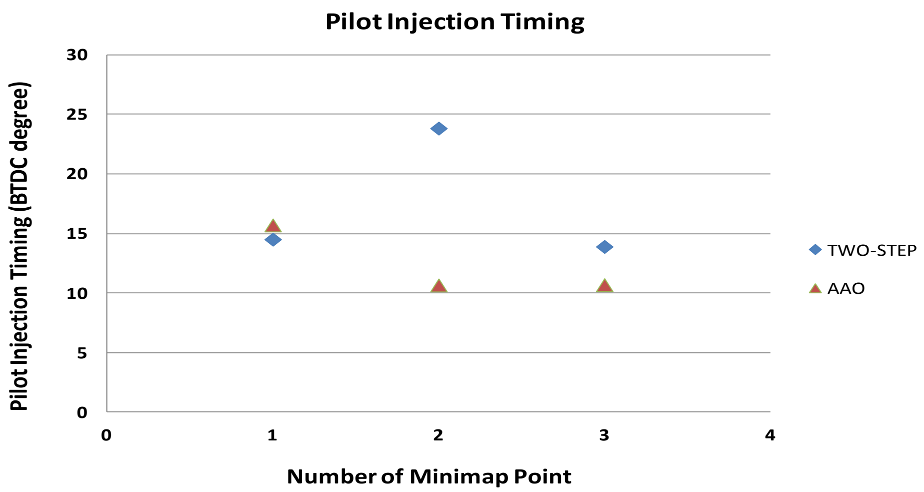


Figure 5.8 Results Analysis (Cycle Emissions): Two-step vs. AAO

In order to understand the effect of engine control variables, Figure 5.9 summarises the results in terms of optimal actuator settings for the 3 minimap points from both the two-step and the AAO optimisation. This shows that main injection timing, pilot injection timing and quantity of pilot injection have quite different actuator settings. The AAO solution makes engineering sense in that it suggests a slightly higher quantity of fuel in the pilot rather than the main injection, which is known to reduce fuel consumption.







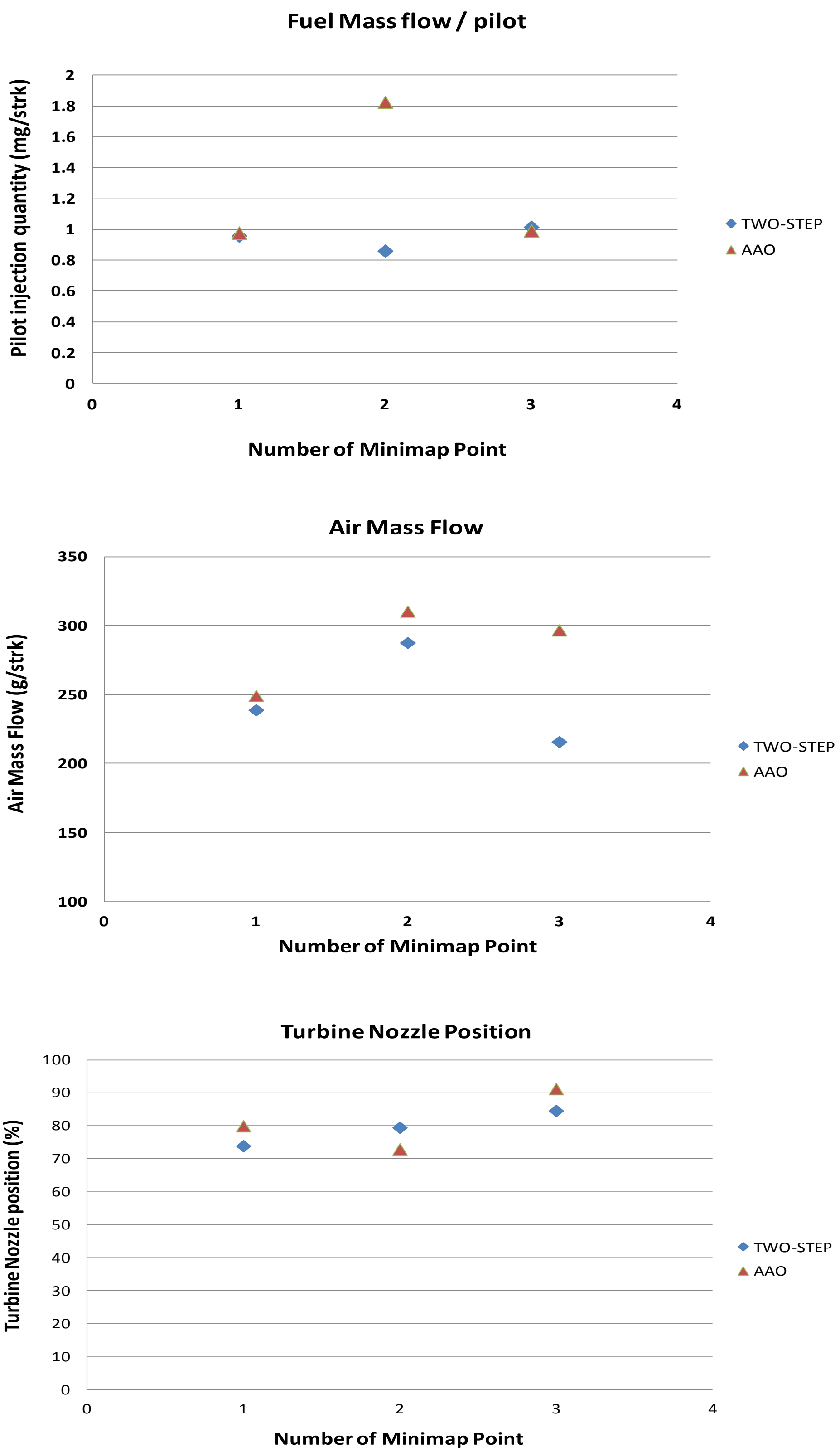


Figure 5.9 The Actuator Settings of Optimal Solutions

5.5 Preliminary AAO Implementation over the Whole Drive Cycle:

After the successful implementation for three minimap points, the AAO approach was applied for the whole drive cycle (ten minimap points) optimisation problem. The same data structure design was used in this implementation and the same GA algorithm options were used for the initial test run.

However, after running a few trials, the attempts all failed with a 'no feasible solution found' message, as illustrated in Figure 5.10. Figure 5.10 also shows that the AAO optimisation was terminated after only two GA generations algorithm.

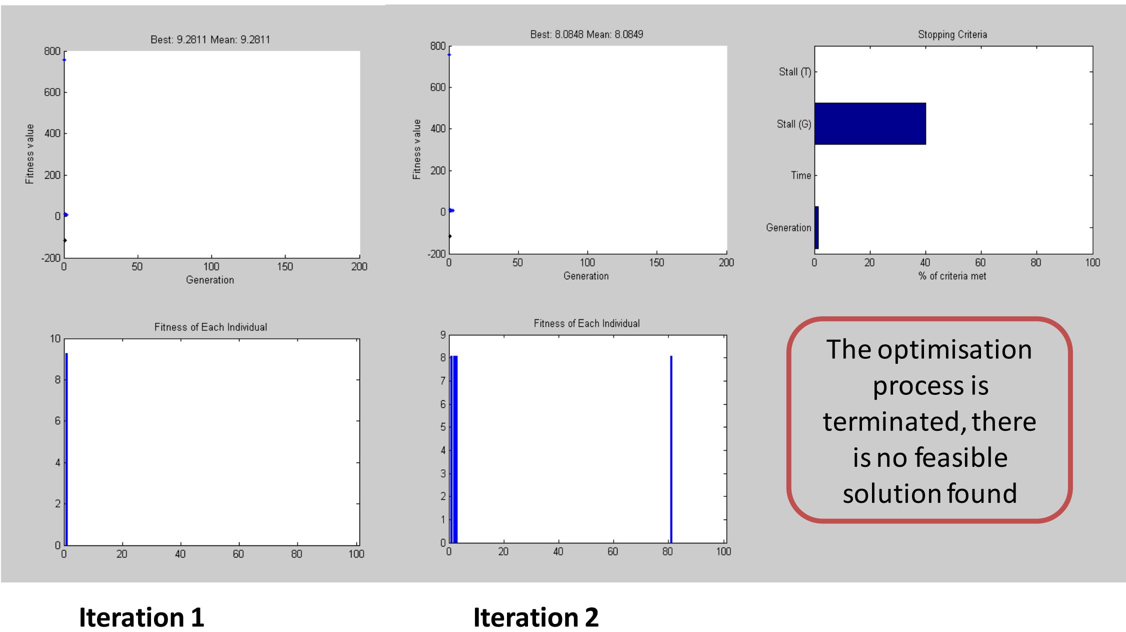


Figure 5.10 AAO Optimisation Approach Plot with Standard GA Algorithm

The plot in Figure 5.10 shows the individual feasible solution objective value from the whole population. From Figure 5.10, it can be seen that there is only one feasible solution in the initial population set. A study using the existing GA algorithm was carried out to understand how the GA algorithm handles this problem. Based on the study of the algorithm implemented in this GA algorithm, Figure 5.11 was developed to illustrate the flowchart of the GA procedure.

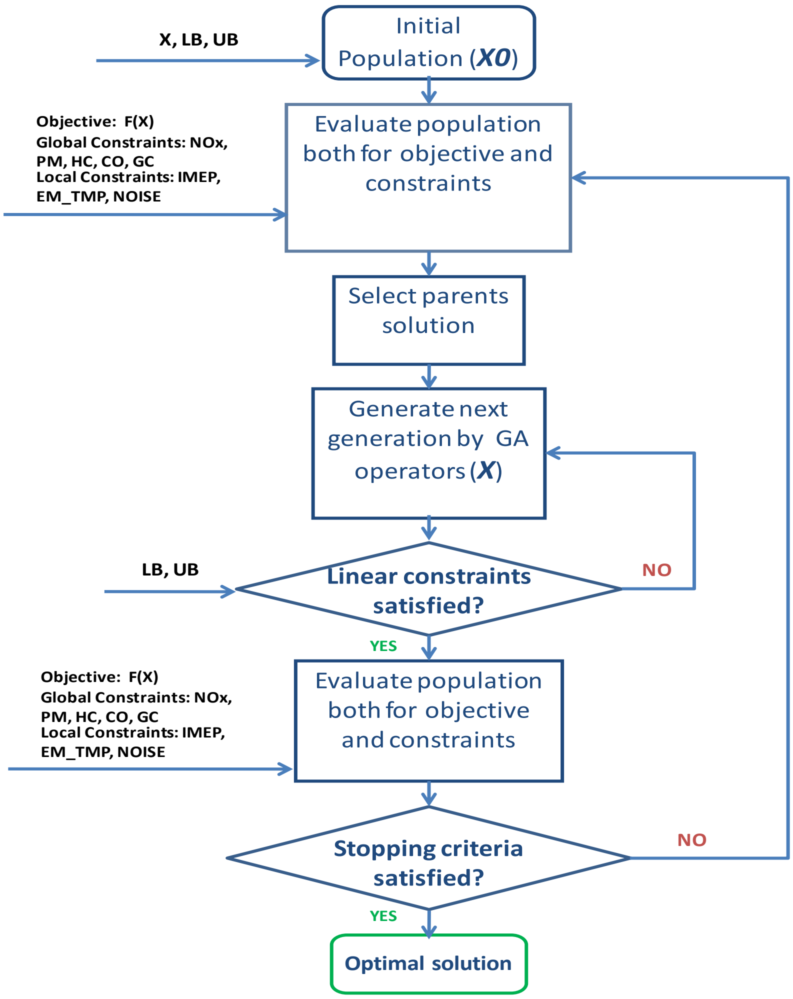


Figure 5.11 Standard GA Procedure Flowchart

As a start, the initial population is generated within all the variable valid ranges. This initial population is passed to the solution evaluator to assess all the solutions in the population. All the feasible solutions are passed to the selection function to choose the best solution as parent solution to generate the next generation. The solution with the best objective value from the whole population will be chosen as only one parent solution, when there is no feasible solution. Because of the same solution evaluator are used from generation to generation, algorithm, which invariably leads to difficulties with finding feasible solutions.

From an engineering point of view this could be explained by the fact that the design space is heavily constrained. In order to address this drawback, two actions have been undertaken, as follows:

1. Carry out a study of the constraints in order to verify that the design space is not over constrained;
2. Revise the GA algorithm such that it can better deal with the heavily constrained optimisation problem.

5.6 Constraints Analysis:

The aim of this analysis was to evaluate how restrictive each of the constraints is individually as well as when they are imposed jointly. It was also aimed to verify that the calibrators' preference for a smooth actuator map and expressed as a series of "actuator change constraints" are not over-resticting the solution space.

5.6.1 Methodology for Constraints Analysis

The approach taken to carry out this analysis used a Monte Carlo process, as outlined below:

1. Generate a set of 100,000 uniformly distributed values within the $n \times k$ dimensional design space (n is the number of nodes; k is the number of variables at each node);
2. Evaluate the success rate with respect to individual global emissions and gradient constraints over the drive cycle;
3. Evaluate the success rate with respect to combinations of global constraints;
4. Carryout a comparative analysis aiming to identify actuator or transition constraints that are relatively harder to satisfy.

A set of MATLAB scripts has been developed to carry out this analysis as shown in APPENDIX I.

5.6.2 Results and Analysis:

5.6.2.1 Evaluation against Individual Constraints:

Table 5.6 summarises the results of checks against individual emissions constraints over the drive cycle. This analysis shows that NOx and PM are the most difficult emission constraints with a success rate of 1% for NOx and 25.5% for PM (the success rate was evaluated as the probability of finding a feasible solution when a specific constraint was imposed, for convenience expressed as a percentage).

This is consistent with reports from the literature, e.g. (Ropke 2009), and the current industry practice of starting the optimisation process by seeking 'local' optimal solutions (i.e. at each given engine speed and load operating point) as a trade-off between NOx and PM, which is widely recognised as being the more difficult to achieve.

Table 5.6 also shows the results from imposing the gradient constraints (GC) checks. This analysis shows that GC is also a very restrictive constraint with a 2% success rate.

Table 5. 6 Individual Emission Constraints and GC Constraints over Whole Drive cycle

Solutions generated: 100,000	NOx	PM	CO	HC	GC
Number of feasible solution	987	25497	96356	99788	1980
Success rate [%]	1.0%	25.5%	96.4%	99.8%	2.0%

A natural subsequent question for this study was to check the distribution of the feasible solutions in the design space, to confirm that feasible solutions are not confined to a corner or a ‘pocket’ of the design space. Given the large number of variables, visualisation was quite difficult, therefore a quantitative measure for the spread of feasible solution was defined in terms of range utilisation for each variable i (Equation 5.3), calculated as the ratio between the observed feasible range (i.e. the range of feasible solution set) and the valid range, i.e. the width of the design space.

$$Range\ utilisation_i = \left[\frac{range\ (feasible\ solution\ set)}{upper\ bound - lower\ bound}\right]_i \qquad \textbf{Equation 5.3}$$

Figure 5.12 shows the distribution of range utilizations for each variable for the feasible solutions against the NOx constraint; the analysis of this result shows high range utilization for nearly all variables.

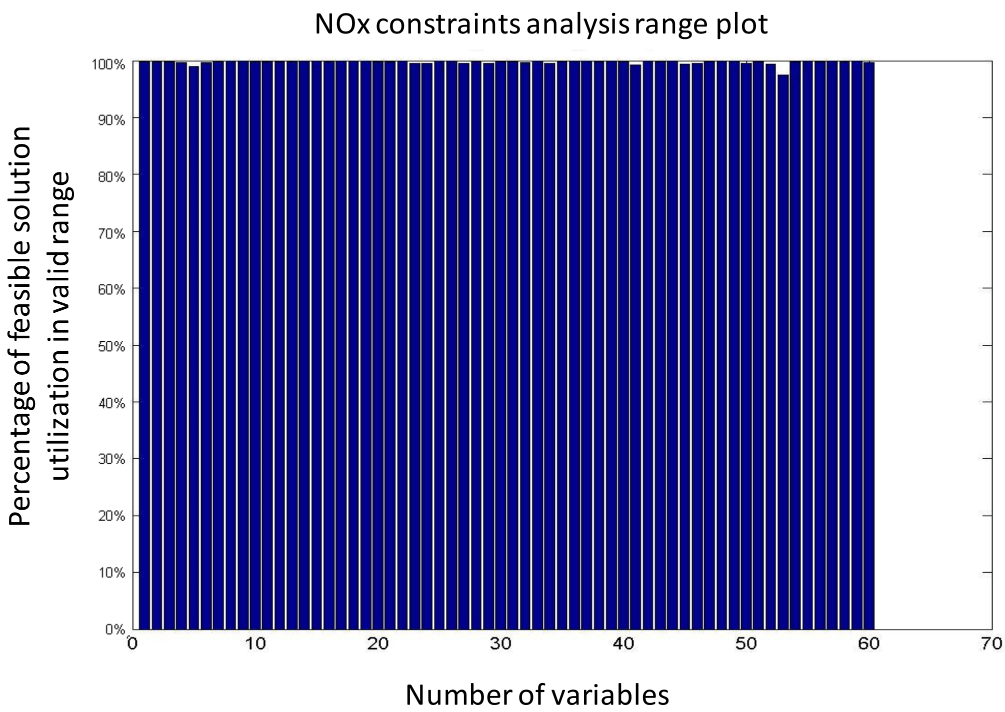


Figure 5.12 Chart of range utilization against NOx emission constraint

A similar analysis was carried out for the Gradient Constraints (GC), with the result illustrated in Figure 5.13. This analysis shows that most actuators utilize the design space well, apart from variable pilot quantity at node1, and variable fuel pressure at node 7.

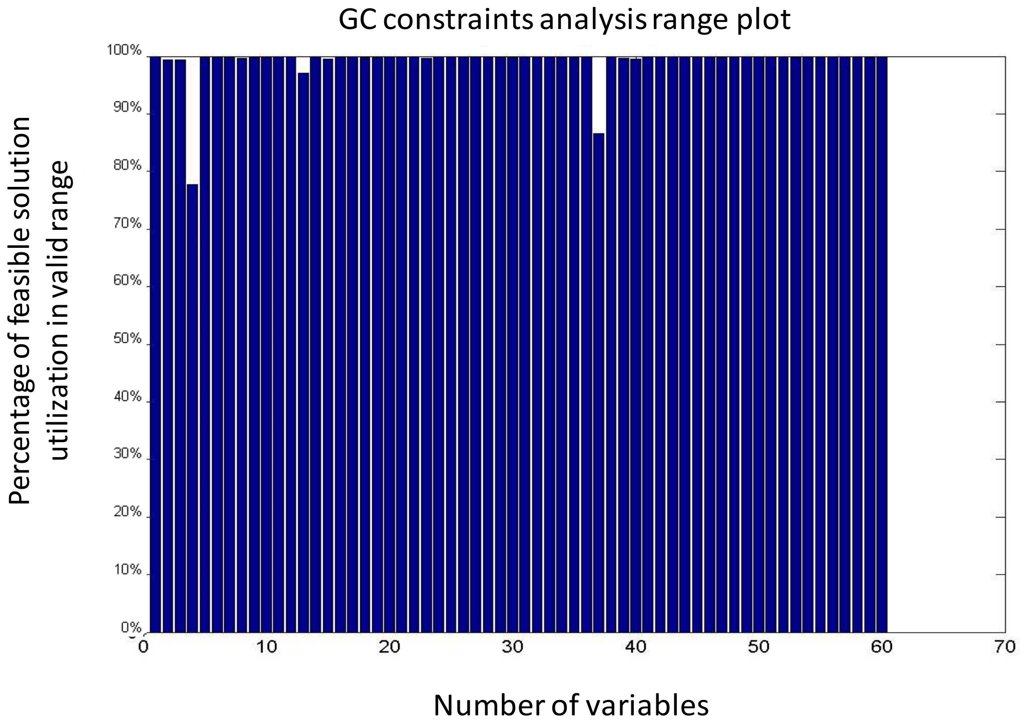


Figure 5.13 Bar Chart of range utilization against gradient constraints

5.6.2.2. Evaluation for Combinations of Constraints

Table 5.7 summarizes the results of a number of simulations against the different constraint combinations. The analysis shows that the combinations of constraints are even more restricting than those calculated from individual constraints analysis. This analysis shows that when NOx and PM constraints are jointly imposed the success rate is 0.042%. It was also found that when all constraints are simultaneously imposed, there was no feasible solution in the randomly generated population.

Table 5. 7 The different combinations of emission constraints over drive cycle

Solutions generated: 100,000	NOx-PM	NOx-GC	PM-GC	All emission constraints	All constraints
Number of feasible solutions	42	23	403	32	0
Sampling Success rate [%]	0.042%	0.023%	0.403%	0.032%	0

A range utilization plot for the NOx-Pm combination of constraints is shown in Figure 5.14.

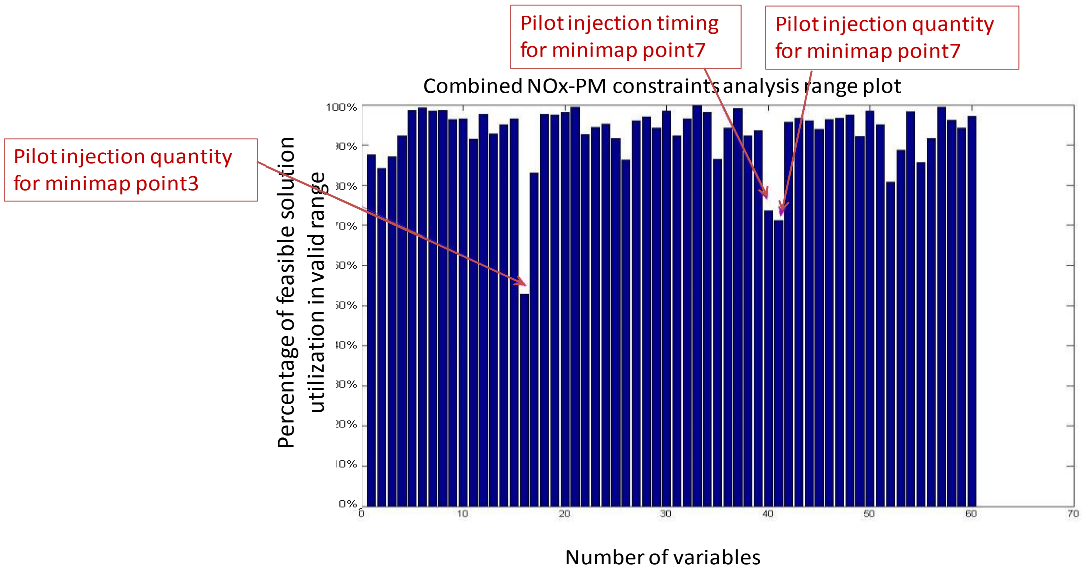


Figure 5.14 Chart of Range Utilization against NOx-PM Emission Combination Constraints

Figure 5.14 shows that in particular the pilot injection quantity at node3 was reduced to about 50 % of the valid range and pilot injection timing and quantity at node 7 was reduced by about 30 % of the valid range.

Figure 5.15 shows that the Mass Air Flow (MAF) at minimap point 6 was reduced to about 40% of the valid range and pilot injection at minimap point3 was reduced by more than 50% of the valid range, when all emission constraints are imposed.

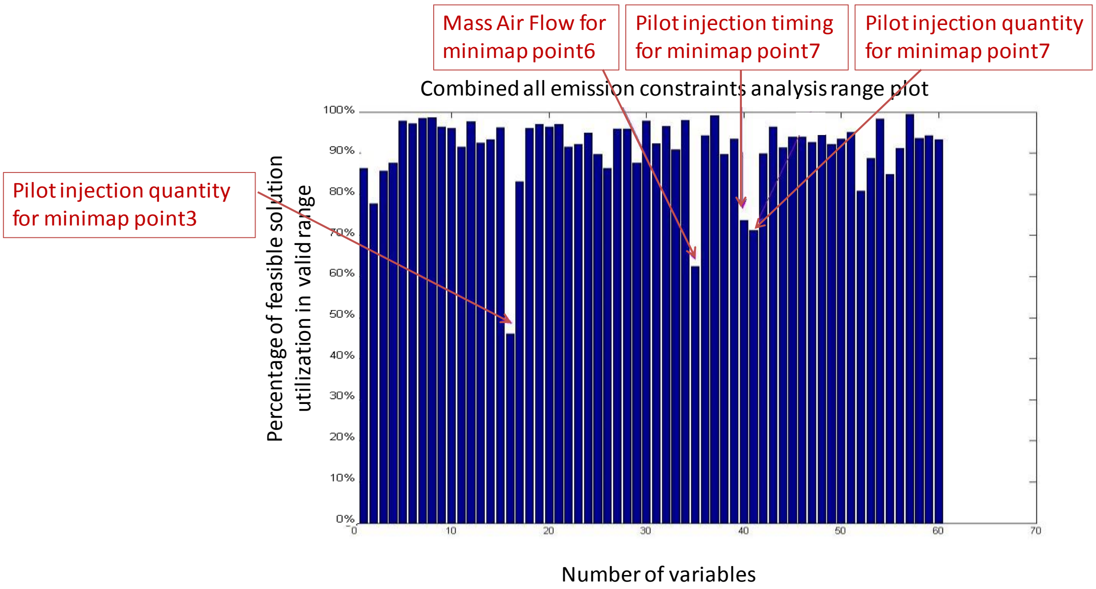


Figure 5.15 Bar Chart of range utilization against all cycle emission combination constraints

Figure 5.16 shows that when the most difficult constraints (i.e. NOx and GC) are simultaneously imposed a number of variables have been reduced by about 50% of the valid range viz. main injection timing and pilot injection quantity at node1; fuel pressure and MAF at node 7; and MAF at node9.

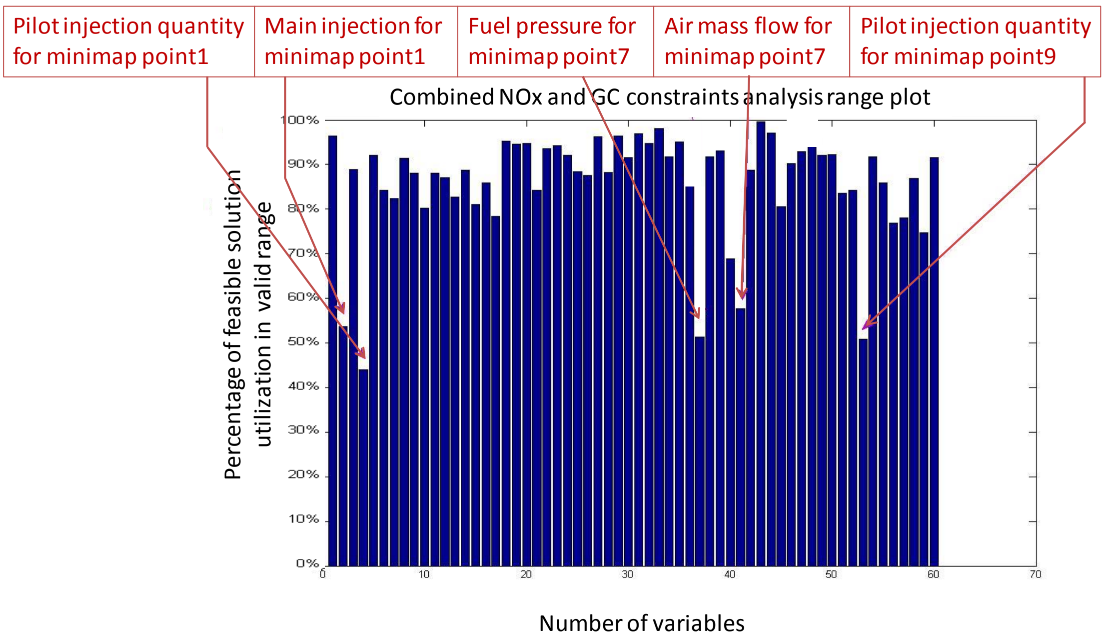


Figure 5.16 Chart of range utilization against the combination of NOx-GC constraints

Further analysis was carried out to evaluate the relative difficulty of gradient constraints for particular variables or transitions. The methodology for this analysis was to take the viable solutions after imposing all the emissions constraints, and impose actuator change trasitions in turn to evaluate success rate. From a mathematical point of view this can be stated as evaluating the conditional probability of success against a specific gradient constraint given that the emissions constraints are satisfied.

Given that the number of feasible solutions for all emissions constraints are very small, the Monte-Carlo experiment was repeated a further 2 times, which yielded 85 emissions feasible solutions (out of a simulated population of 300,000). Each of these 85 solutions was tested in turn against each of the actuator change. Table 5.8 summarises this analysis in terms of how many times a particular gradient transition constraint was infringed (out of the total of 85). The transitions are shown in Table 5.8 with respect to the minimap reference number for the start and stop point. The apparent most difficult minimap transitions and actuator variable have been highlighted in Table 5.8. Figure 5.17 summarizes the Gradient Constraints analysis in graphical terms, highlighting the most difficult transitions in the speed-load space.

Table 5. 8 Gradient Constraints Failing Statistic

		Actuators Gradient Constraints Failed Times						
Start point	stop point	Fup	soi_main1	soi_prev2	mf_prev2	maf	bpa	Average of failing rate
1	2	18%	12%	11%	31%	0%	13%	14%
1	3	21%	19%	14%	41%	0%	2%	16%
1	7	33%	9%	11%	31%	0%	0%	14%
2	3	24%	31%	14%	53%	2%	1%	21%
2	6	13%	15%	12%	27%	13%	6%	14%
2	7	18%	21%	13%	29%	0%	7%	15%
3	4	19%	18%	26%	33%	15%	9%	20%
3	5	11%	21%	13%	31%	0%	1%	13%
3	6	46%	21%	12%	36%	11%	0%	21%
3	7	42%	14%	29%	55%	0%	8%	25%
3	9	14%	9%	16%	26%	8%	7%	13%
4	7	22%	11%	18%	39%	0%	2%	15%
4	9	25%	12%	15%	42%	6%	9%	18%
4	10	18%	6%	12%	27%	0%	6%	12%
5	6	14%	5%	13%	44%	11%	2%	15%
5	8	12%	18%	11%	34%	0%	5%	13%
5	9	44%	16%	32%	35%	1%	8%	23%
5	10	24%	14%	22%	33%	19%	6%	20%
6	9	22%	5%	14%	42%	2%	0%	14%
8	10	28%	7%	9%	31%	1%	1%	13%
9	10	39%	7%	20%	26%	4%	7%	17%
Average of failing rate		24%	14%	16%	36%	4%	5%	

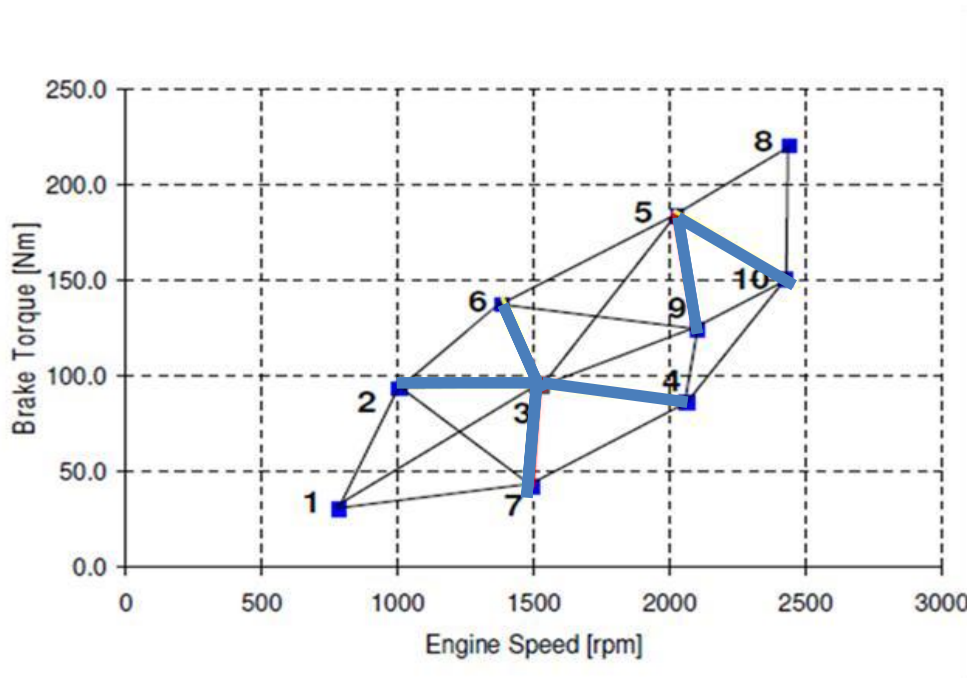


Figure 5.17 Transitions over the drive cycle

The transitions higlighted in Figure 5.17 (i.e. transitions from minimap points 3 to 7, and 9 to 5) were found to be the most difficult. This makes engineering sense as they are associated with a sharp increase in engine load. Therefore, some of the GC constraints are possibly too restrictive to satisfied for the transitions with sharp increase engine load. Figure 5.17 also shows that the transitions around minimap point 3 were more difficult than the others. The statistics shown in Table 5.8 also concluded that the most difficult actuators to satisfy the gradient change constraints are “pilot injection quantity” (variable code *mf_prev2*) and fuel pressure (variable name *Fup*).

5.7 Development of an Improved GA Creation Function

The constraints study has highlighted the difficulty of finding feasible solutions, which explains why the standard existing GA tool failed to generate a feasible initial population and the optimisation ends up prematurely with ‘no feasible solution found’.

In order to address this issue within the GA algorithm, a mechanism was designed to bring the infeasible solution into the feasible area by using a subsidiary optimisation algorithm. Figure 5.18.a illustrates this idea, which is to find the closest feasible solution given the initial un-feasible “guess” solution. Thus, if X0 (in Figure 5.18a) is the starting point for the optimisation procedure, the searching direction is forced towards the closest feasible point X by minimising the distance from X0 to X. In generating the initial population, the diversity also is an important feature for the GA process in terms of finding the global optimal solutions. Therefore, the mechanism is required to not only generate feasible solution, but also keep the diversity. Figure 5.18b demonstrates the process to generate feasible population based on a factorial design in two dimensions. Figure 5.18b illustrates the idea of a feasible population design which preserves diversity within a nonlinear constrained design space, underpinned by a factorial designed experiment to generate the initial guess solutions. In Figure 5.18b, the solid dots are infeasible solutions which are locate outside of feasible area, and the arrows show the movements from infeasible solution to the nearest feasible area.

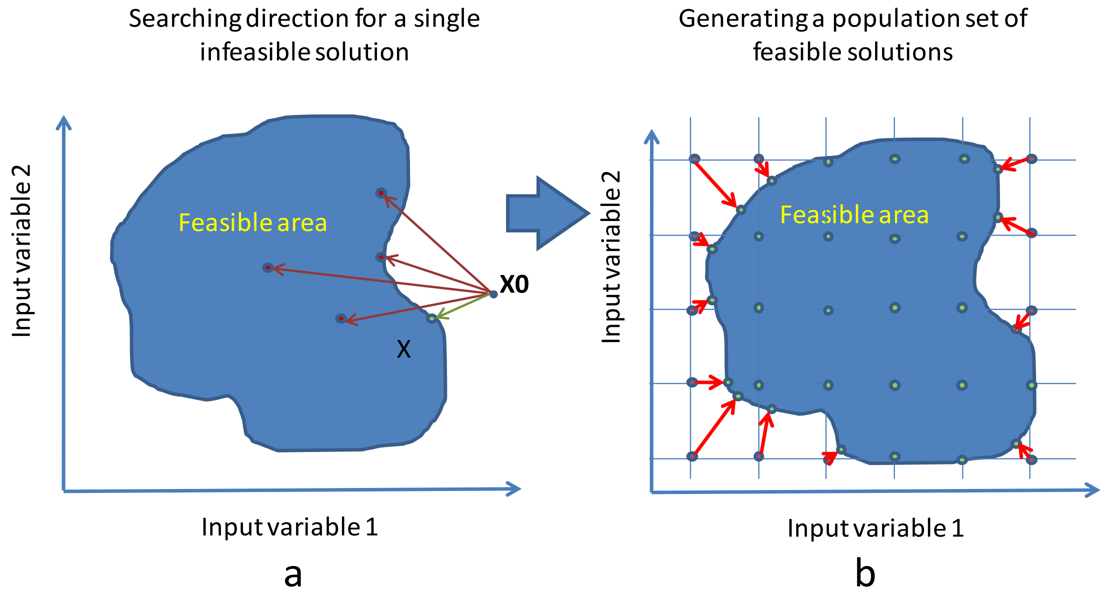


Figure 5.188 Illustration of Creating Initial Feasible Population by Bringing the Infeasible Solution into the Closest Area

The formulation of the optimisation problem is shown in Equation 5.3.

Min: $D(X) = \sum(X - X0)^2$

ST: $LB < X < UB$

$$G(X) = \sum_{i=1}^{n} w_i * Emission_i < Emission\_Limit$$

$$H(X)_i = Local\ \_constraint_i < Local\_constraint\_limit_i$$

$$GC(X_i, X_j) = (X_i - X_j) < GC\_limit_{ij} \qquad i \neq j, \{i,j\} \in (1,2, \ldots n)$$

Equation 5.3

A Matlab script was created to implement this procedure, which was called "corrector", shown in APPENDIX V. Figure 5.19 shows the flowchart of the feasible initial population creation function. This newly designed initial creation function employs an optimal space filling DoE method to generate an initial population within the valid range, which is a strategy in order to ensure a good spread (directly linked to diversity) in the solutions space. Solutions (GA individuals) are saved in the feasible population if they satisfy all the nonlinear constraints; otherwise, they are sent to the "corrector" function to find the closest feasible solution, as illustrated by the flowchart in Figure 5.19.

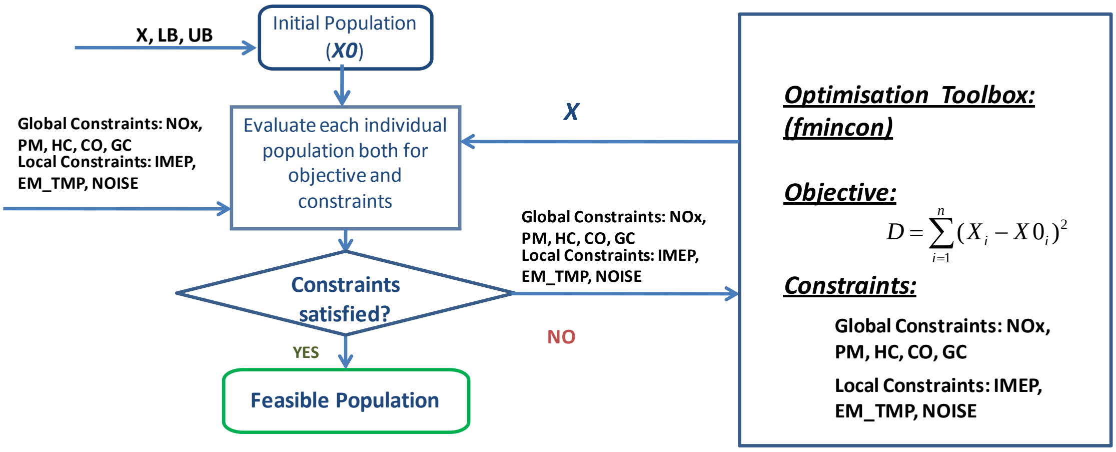


Figure 5.19Flowchart of the Feasible Initial Population Creator

5.8 AAO Results with Corrected GA Creation Function

5.8.1. Initial Results and Analysis

The AAO optimisation was run with the modified creation function for the whole drive cycle (ten minimap points). Figure 5.20 illustrates the convergence plot of the GA process, which still shows that a significant number of solutions in the population are lost as infeasible.

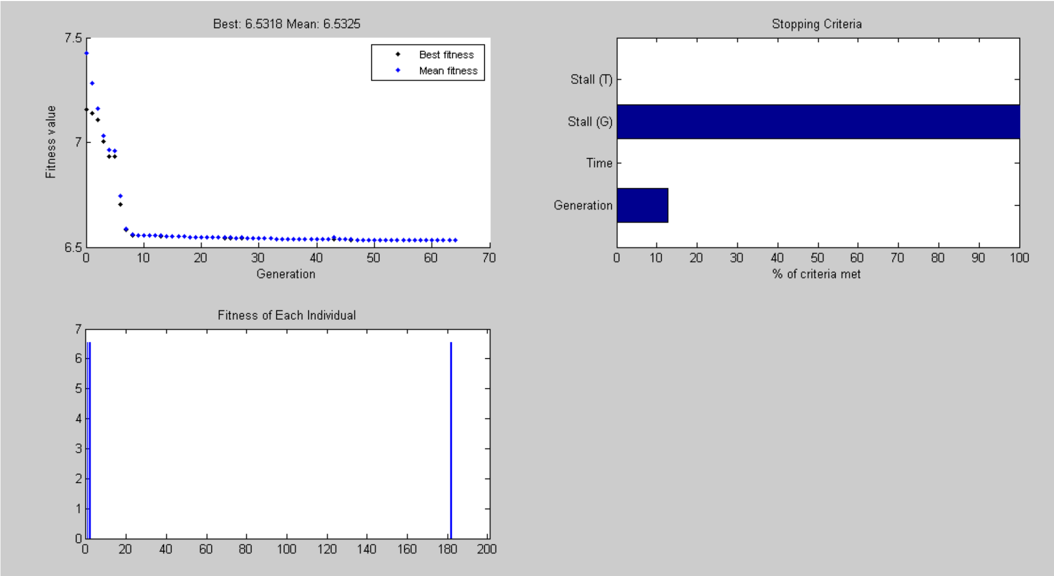


Figure 5.19 Failed AAO Test Convergence Plot

The loss of feasible solutions with subsequent GA iterations results in premature convergence (less than 15% of stall generation limits). The use of "elite" option guaranteed that at least a small number of feasible solutions are preserved. Figure 5.21 illustrates the improvement in the overall objective (fuel economy) from the AAO approach compared with the two-step approach.

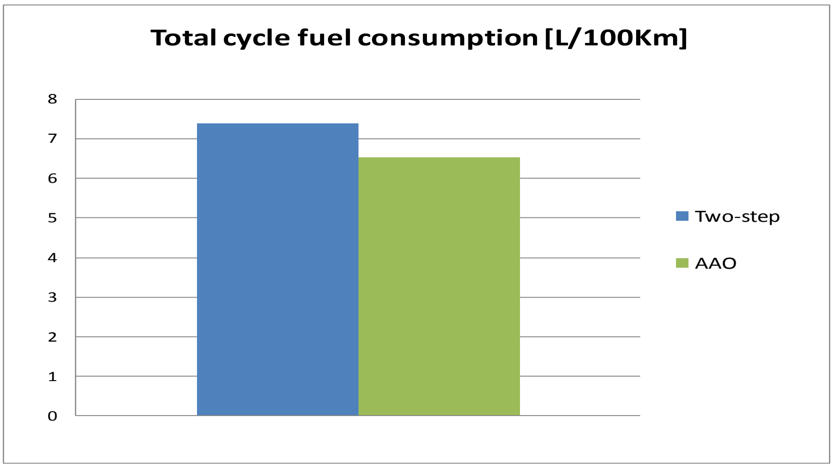
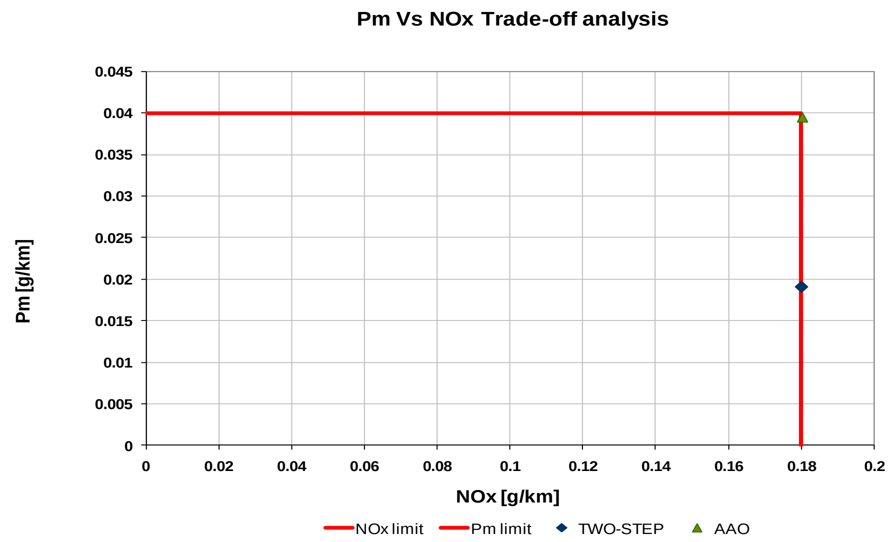


Figure 5.21 Fuel Comparison of between Two Step Approach and AAO Approach with Feasible Initial Population

Figure 5.22 shows the trade-off plots for both CO-HC and PM-NOx, showing that the emissions constraints have been met.



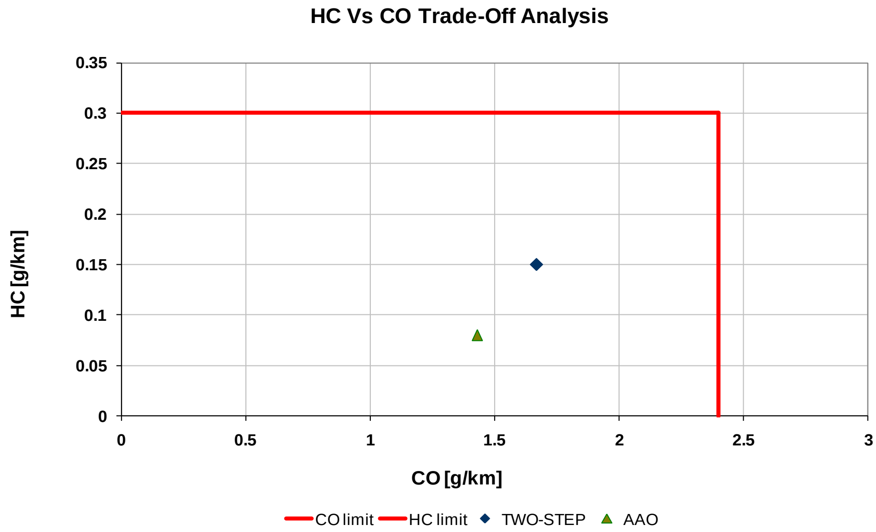


Figure 5.20 Pm vs. NOx and HC vs. CO Emissions Comparison of between Two Step Approach and AAO Approach with Feasible Initial Population

5.8.2. Further Study of the GA Algorithm Performance

In order to understand the optimisation process and improve the quality of optimisation process, a study was taken based on the optimisation test with the feasible initial population, which has been generated by the creation function. The optimisation test was shown the early convergence, which was illustrated in figure

5.21. Figure 5.23 shows the first 2 iterations of the optimisation test. It can be seen that a significant number of feasible solution are missing from the first 2 generations.

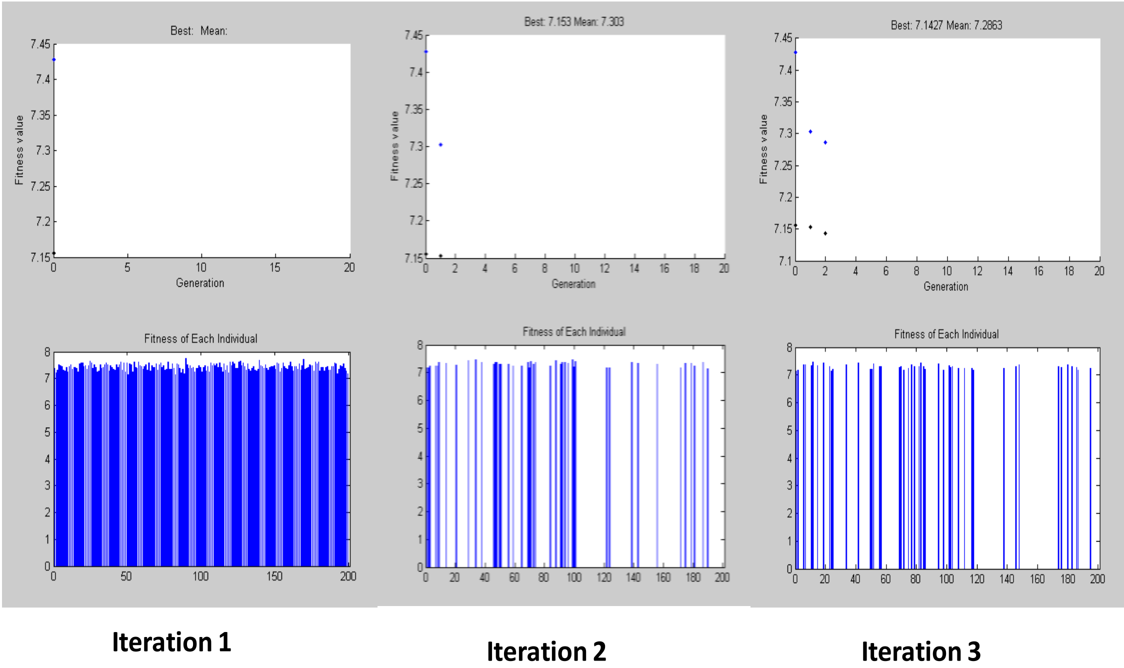


Figure 5.21 Feasible Population Plot from Iteration1 to Iteration3

The analysis in Figures 5.20 and 5.23 suggest that the number of feasible solutions in the population is decreasing sharply from iteration to iteration. It is seen from Figure 5.20 that only three feasible solutions were left in the last population when the optimisation process converged (which includes the “elite” option). It is clear that the loss of feasible solutions from generation to generation may cause the early convergence and lack of robustness in the optimisation process.

5.8.3 Development of Revised GA Operators

In order to address this issue of the loss of feasible solutions, the same idea of a ‘corrector’ algorithm can be employed for the GA operators to ensure that the solutions in the next generation are feasible. This ‘Corrector’ function is embedded into newly developed operators, which are named ‘XFcrossover’ and ‘XFmutation’. Both ‘XFcrossover’ and ‘XFmutation’ operators were developed based on the standard GA operators by adding the ‘Corrector’ function, in order to bring the infeasible solution into feasible area. Figure 5.24 illustrates the use of ‘corrector’ function in the existing GA algorithm.

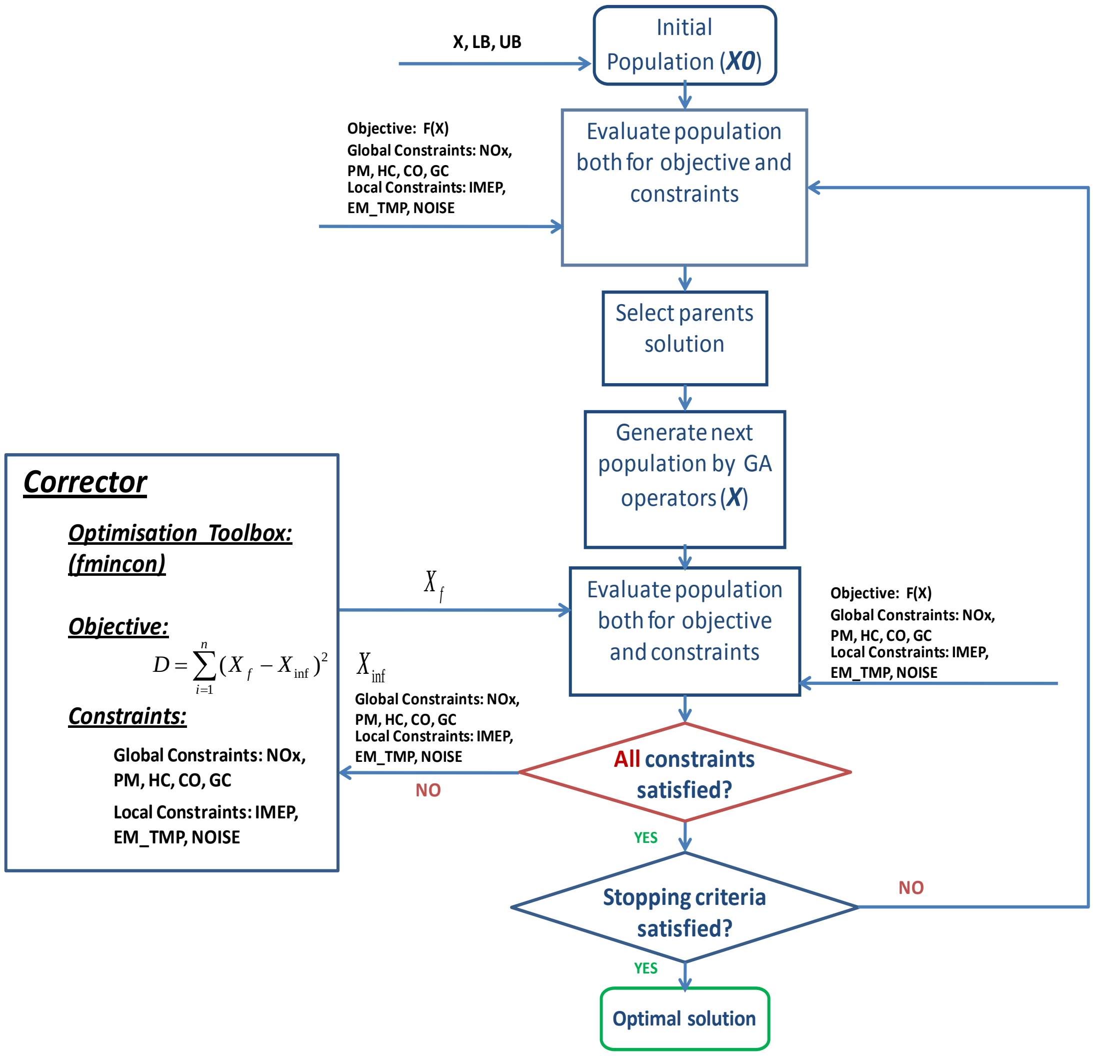


Figure 5.22 Illustration of AAO Flowchart Applied Genetic Algorithm with Modified Operators

5.8.4. AAO Implementation with Modified GA Operators and Results Discussion

Table 5.9 summarises the GA options for AAO implementation with modified operator functions. The standard operators have been modified and chosen by setting up the options for the GA optimisation algorithm.

Table 5. 9 GA option settings for AAO test with modified GA operators

Option name	Setting
Selection Fcn	selectiontournament,2
MutationFcn	XFmutation
CrossoverFcn	XFcrossover
Generations	20
Stall Gen Limit	4
Elite count	2
Crossover fraction	0.7
TolCon	0.01
TolFun	0.01

Figure 5.25 shows the implementation of the AAO approach program structure.

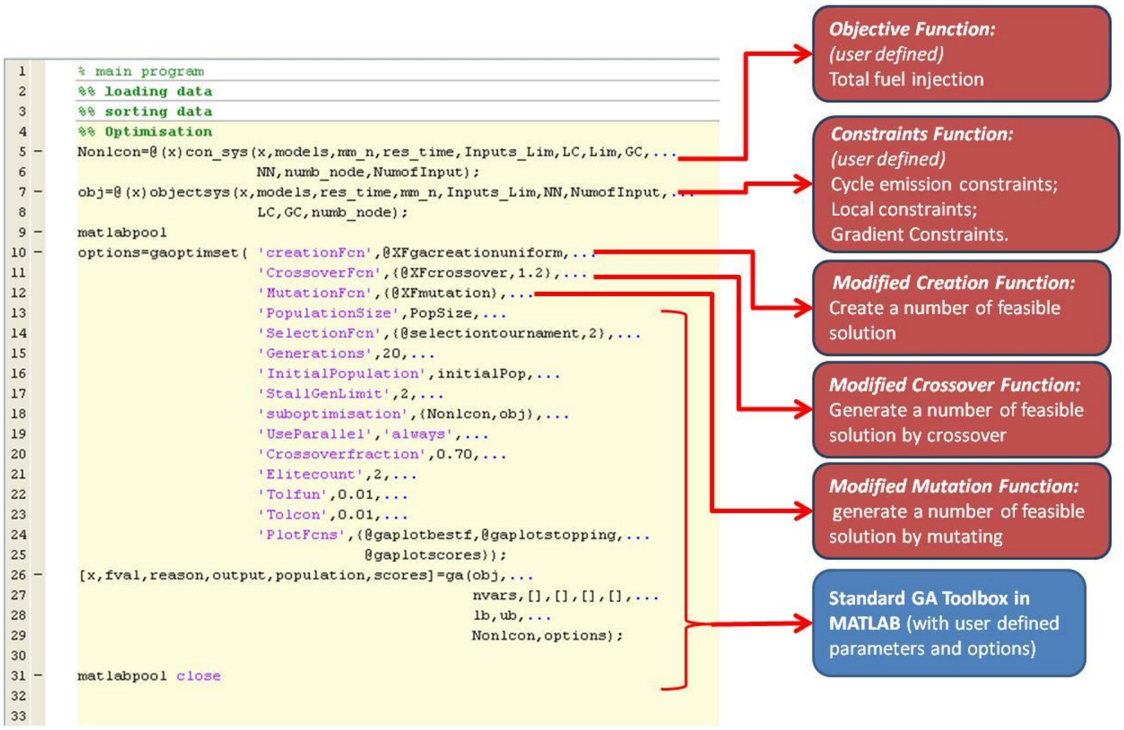


Figure 5.23 Illustration of AAO Approach Implementation with modified operators and function

The AAO optimisation test was carried out with ten nodes using modified crossover and mutation operators and the creation function to run the test. Figure 5.26 shows the GA convergence of the AAO approach and the smoothness of the final result. The optimisation process stopped after 18 generations when it reached the ‘stall’ generation limits. The fuel consumption was reduced to about 6.21L/100km which represents an improvement of 15% compared with the two-step approach, as illustrated in Figure 5.27.

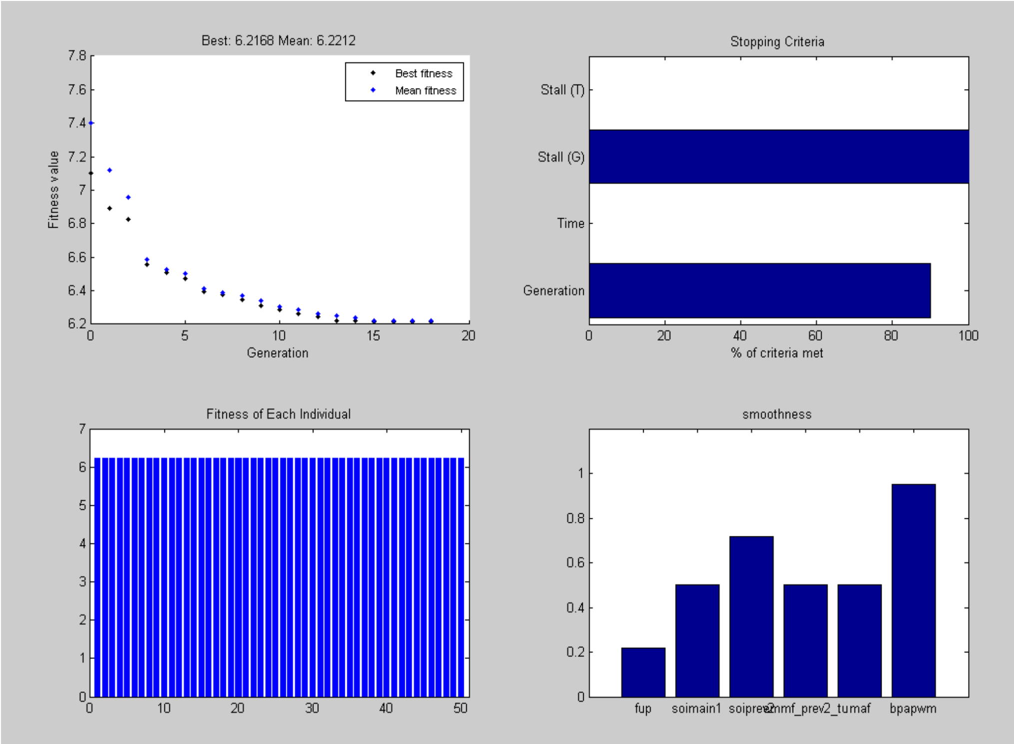


Figure 5.24 All At Once approach testing results with 10 nodes by applying modified GA algorithm

Figures 5.28 and 5.29 show that the emissions performance of the AAO solution, incomparison to the “current” 2-stage process solution. It is clear that all emissions are within the set engineering limits, however, the AAO solution is clearly much better in terms of fuel economy.

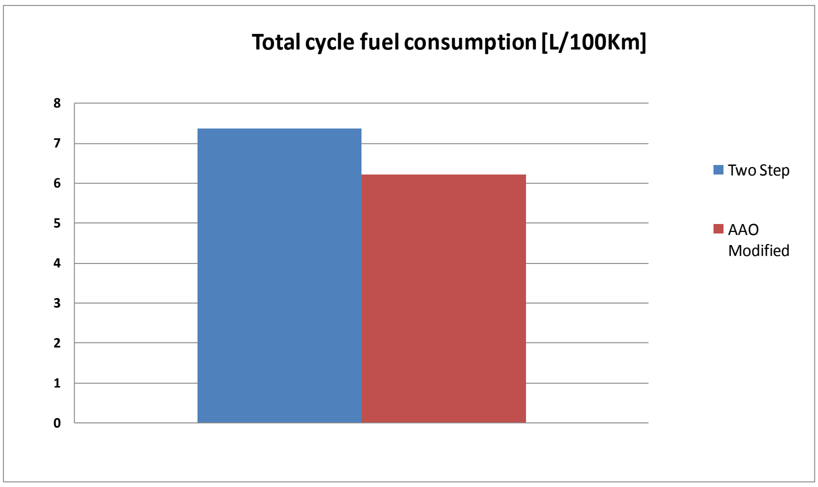


Figure 5.25 Comparison of Fuel Consumption

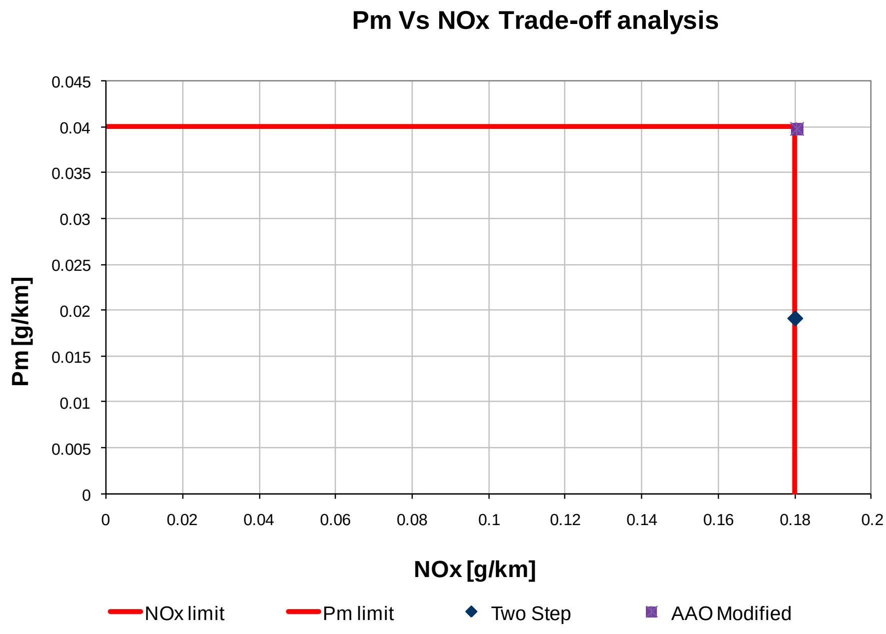


Figure 5.26 PM-NOx Emission Plot with Engineering Targets

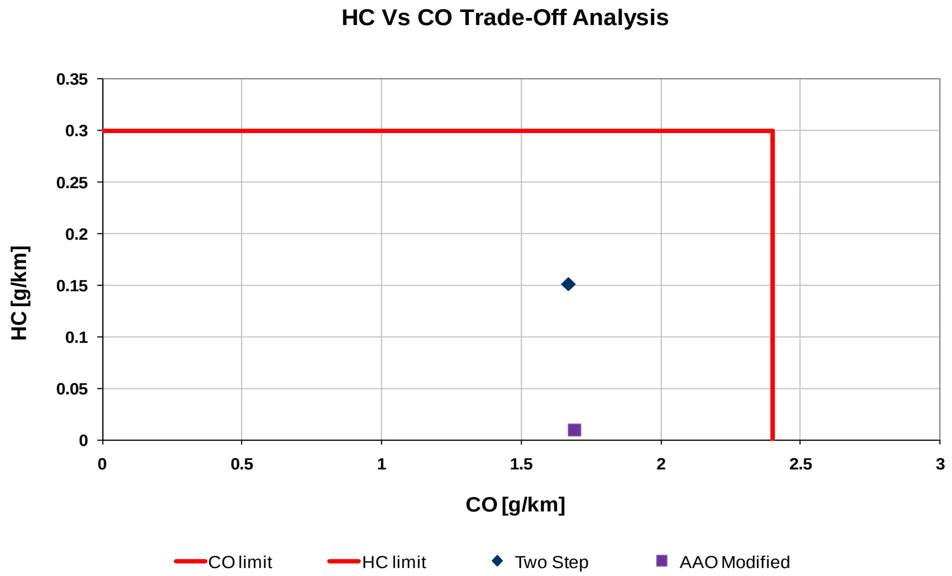
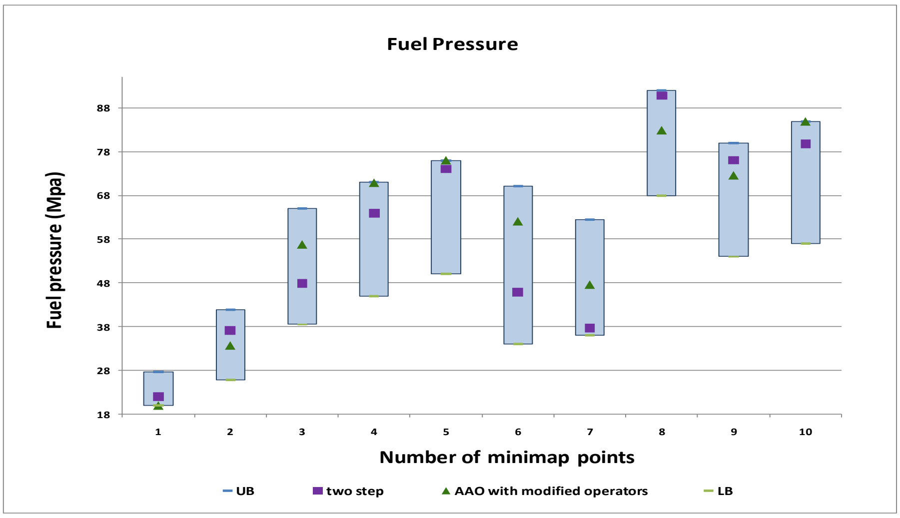
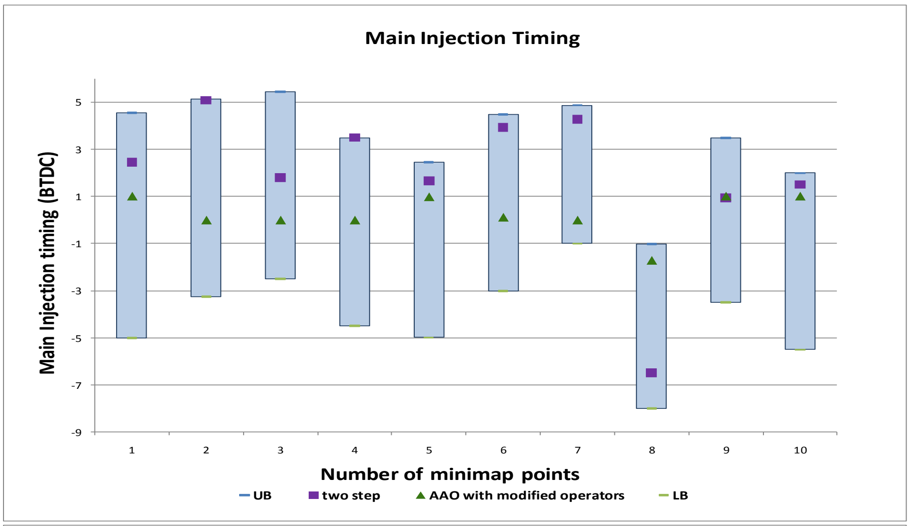
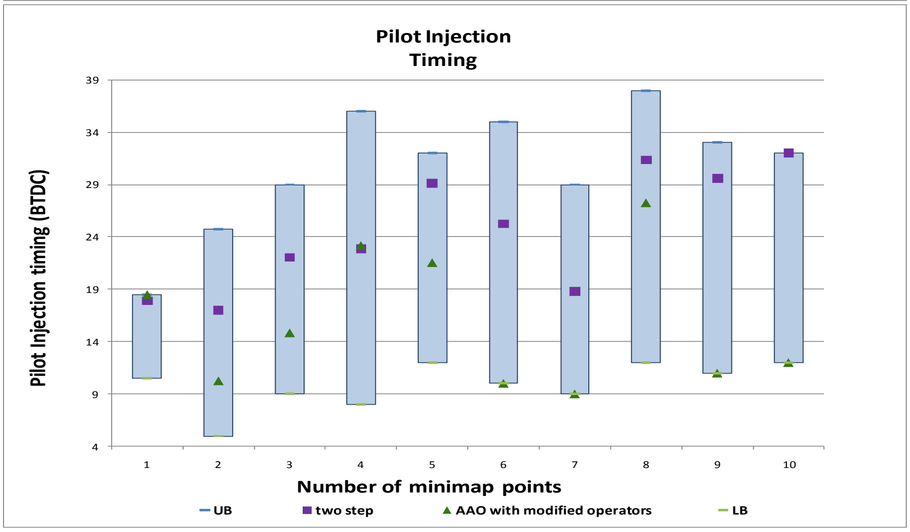


Figure 5.27 HC-CO Emission Plot within the Engineering Limits

The graphs in Figure 5.30 show the distribution of optimal solutions in the valid actuators range.


Fuel Pressure
Fuel pressure (Mpa)
88
78
68
58
48
38
28
18
1
2
3
4
5
6
7
8
9
10
Number of minimap points
UB
two step
AAO with modified operators
LB


Main Injection Timing
Main Injection timing (BTDC)
5
3
1
-1
-3
-5
-7
-9
1
2
3
4
5
6
7
8
9
10
Number of minimap points
UB
two step
AAO with modified operators
LB


Pilot Injection
Timing
Pilot Injection timing (BTDC)
39
34
29
24
19
14
9
4
1
2
3
4
5
6
7
8
9
10
Number of minimap points
UB
two step
AAO with modified operators
LB

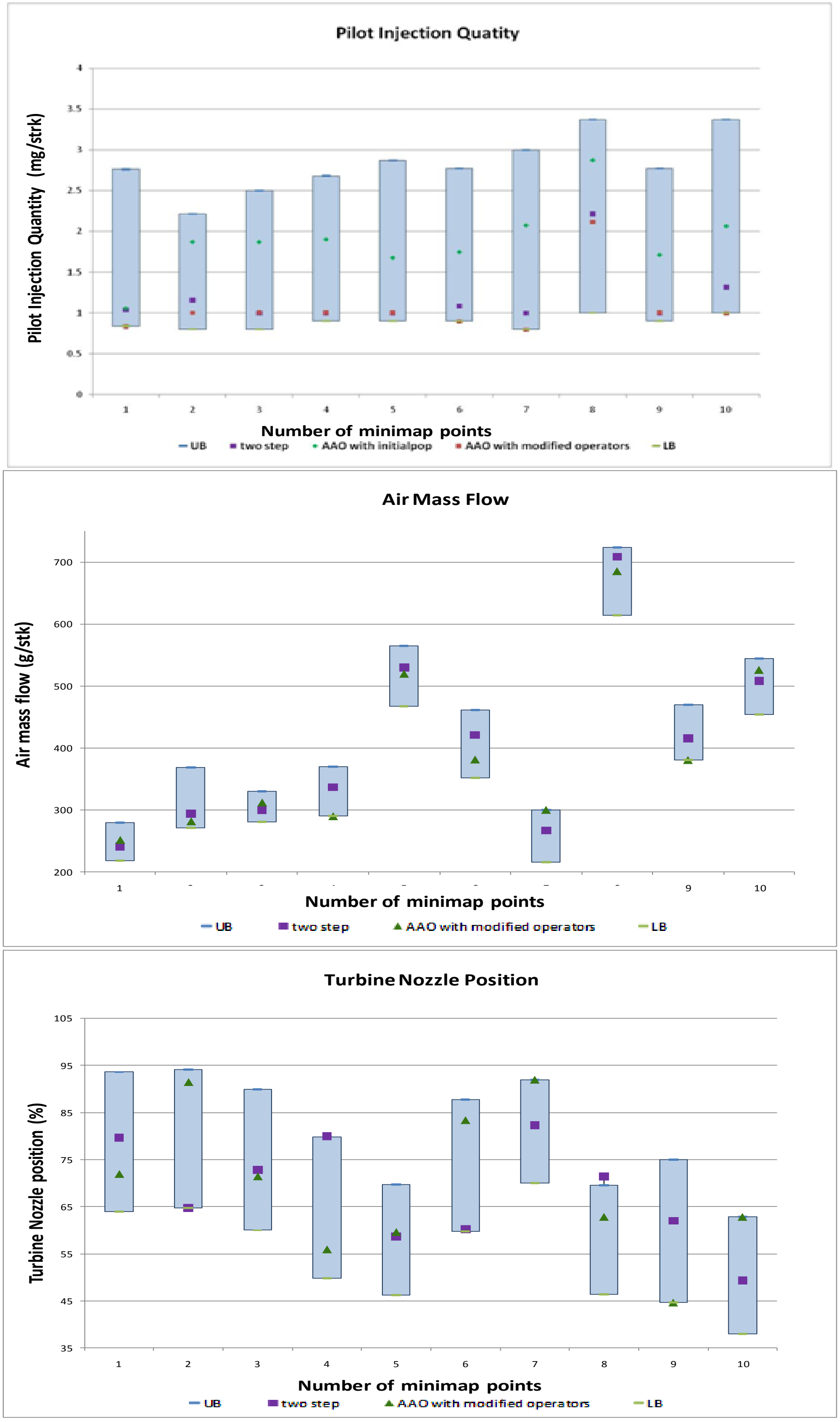


Figure 5.28 Plot of different optimal solution of actuator settings in valid range

The graphs in Figure 5.30 show that the actuator settings of different optimal solutions are spread over the valid range. The optimal solutions from both the AAO and the two-step approaches are located in different areas for some actuators, especially the main injection and pilot injection timing.

In terms of computation the AAO approach only completes the optimisation within less than 10 hours by using the standard GA algorithm, while the AAO approach with modified operators used about 60 hours with the smaller population size of 50. Therefore, compared with the two-step approach (more than few days); both AAO approaches applying standard GA algorithm have the benefit in terms of computation.

In order to judge the effectiveness and efficiency of this AAO problem formulation, it was important to know whether the formulation kept the computational effort manageable. With the three minimap points' application, the AAO approach not only improved the fuel consumption, but also performed better computation efficiency than the current method in Diesel engine calibration process. Nevertheless, the test for the full engine test drive cycle optimisation showed the difficulties of generating feasible solutions efficiently in a heavily constrained space. In order to address this problem, special algorithms were developed and written into MATLAB functions for generating feasible solutions for the initial population, as well as offspring from generation to generation via mutation and crossover operators. Consequently, the computation has been slowed down to more than 60 hours for full cycle optimisation test by the modified GA operators function.

5.7 Summary

In this chapter a single level optimisation structure AAO has been developed, tested and discussed. The Diesel engine calibration problem has been formulated into the original AAO structure and tested. The original AAO has shown some advantages in terms of finding the optimal solution by focusing on the objective with respect to all the constraints.

A solution space study has been carried out and shows that the solution space is very heavily constrained. Such a constrained space caused the difficulty of generating feasible solutions and introducing a premature termination and hence an unsuccessful optimisation process. In order to avoid the risk of early convergence, some special functions have been created to help the standard GA algorithm find feasible solutions.

Finally, the AAO optimisation structure was successfully tested with a Diesel engine full drive cycle calibration problem. The result has shown that AAO formulation is capable of finding a global solution and can guarantee a solution for every single run. However, the AAO approach with modified function costs very expensive computation.

Chapter 6 Collaborative Optimisation Approach to Diesel Calibration Optimisation

6.1 Introduction

As discussed previously, the Diesel engine calibration problem is hierarchically structured at two levels, which makes MDO frameworks attractive for this optimisation study. After the evaluation of a number of MDO frameworks (Allison et al., 2009 , Braun and Moore, 1996 , Kim, 2001 , Kroo, 2004 , Papalambros, 2001 , Sobieszczanski-Sobieski et al., 1998), Collaborative Optimisation (CO) appeared as the closest two level MDO framework formulation, with the ability of dealing strong coupling between disciplines, which is likely to benefit the engine calibration optimisation problem.

From the discussion in Chapters 4 and 5, the Diesel engine calibration problem can be treated as a two-level MDO problem. Within the representation of MDO framework in Chapter 5 (Table 5.1), engine drive cycle outcomes and actuator GC constraints were regarded as system level targets (objective or constraints to achieve). At each minimap, the local constraint analysis is required, such as engine noise, IMEP and exhaust temperature. The AAO framework performed a centralized analysis for both global level targets and local level requirements simultaneously. This requires 316 constraints to be satisfied at the same time, which inherently leads to difficulties with finding feasible solutions.

As discussed in chapter 2 (Section 2.4.2.1), the essential feature of the MDO/CO framework (illustrated in Figure 2.8) is that it decomposes the complex optimisation design problem into two levels, namely the subsystems (or disciplinary) and the system level. The overall system objective is minimised at system level while subject to consistency requirement and system level constraints. Subsystems focus on finding the closest solution to the system coordinates while satisfying local requirements. Such a framework provides more freedom for searching for optimal solutions while maintaining the interdisciplinary compatibility by subsystem design. These features ensure the success of the Collaborative Optimisation framework in practical multidisciplinary optimisation design.

The aim of this Chapter is to present the analysis, development and implementation of a Collaborative Optimisation framework for the Diesel calibration optimisation problem.

For simplicity of notation and to avoid confusion with the use of CO acronym for Carbon Monoxide emissions, the acronym TCO will be used in relation to the conventional implementation of Collaborative Optimisation (TCO - Traditional Collaborative Optimisation).

6.2 Collaborative Optimisation Formulation for Diesel Engine Calibration

Kroo (Kroo, 2004) has discussed an important challenge for large scale MDO, which is decomposing a problem whose components are strongly coupled. Auxiliary variables were suggested to decompose the strong coupling between the disciplines. The system optimiser creates 'copies' of the computed shared variable values and passes these down to the disciplines as target or parameters to perform the coupling. Figure 6.1 illustrates the MDO problem with coupling between disciplines. The coupling is commonly defined as a function of a number of local variables from different disciplines.

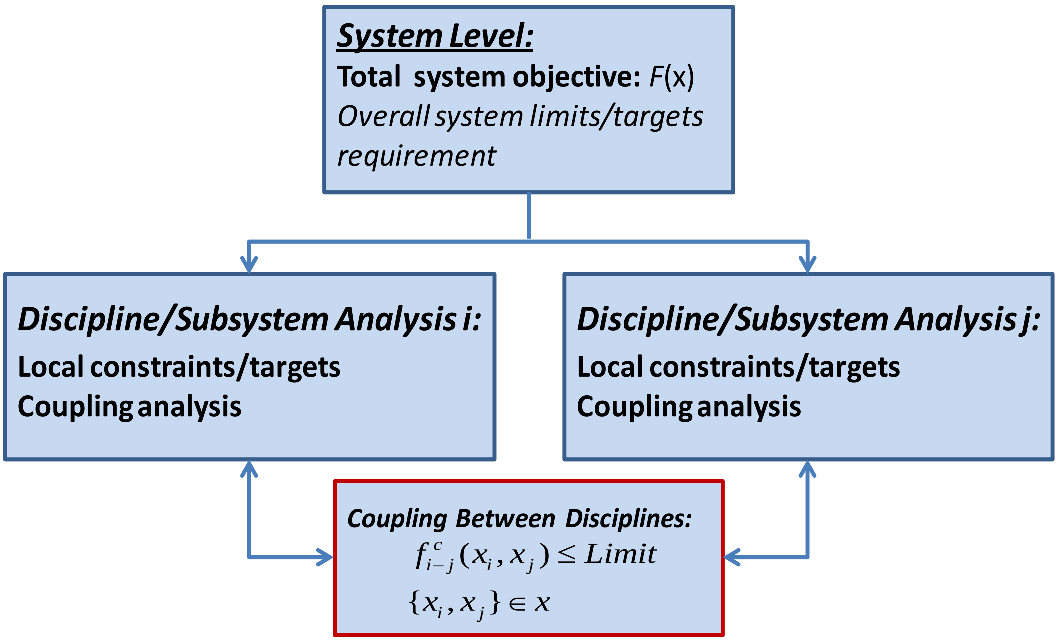


Figure 6.1 Illustration of MDO problem with coupling between disciplines

According to Kroo's suggestion, the strong coupling between disciplines is decomposed by creating auxiliary variables and imposing the coupling conditions at each discipline concurrently. Figure 6.2 illustrates the Collaborative Optimisation principle. Essentially, the coupling between disciplines has not been decomposed, instead, the coupling is performed within individual disciplines in parallel by data communication, and the auxiliary variables are used as medium of the data communication. In Figure 6.2, where *z* is the vector of auxiliary variables, which is used in discipline level optimisation as a target. While the disciplines are strong coupled, these auxiliary variables are also used to perform the coupling between the disciplines. For example, the coupled disciplines in Diesel engine calibration problem is described as $x_i - x_j < limit$, it can be performed at discipline level by using the auxiliary variables $z_i^{sub} - z_j < limit$, where refer to the local constraints evaluation $f_i^c(z_i^{sub}, z_j) < Limit$, the z_i^{sub} is local variables at the discipline *I* and z_j is the artificial auxiliary variable for discipline j, which is created at system level.

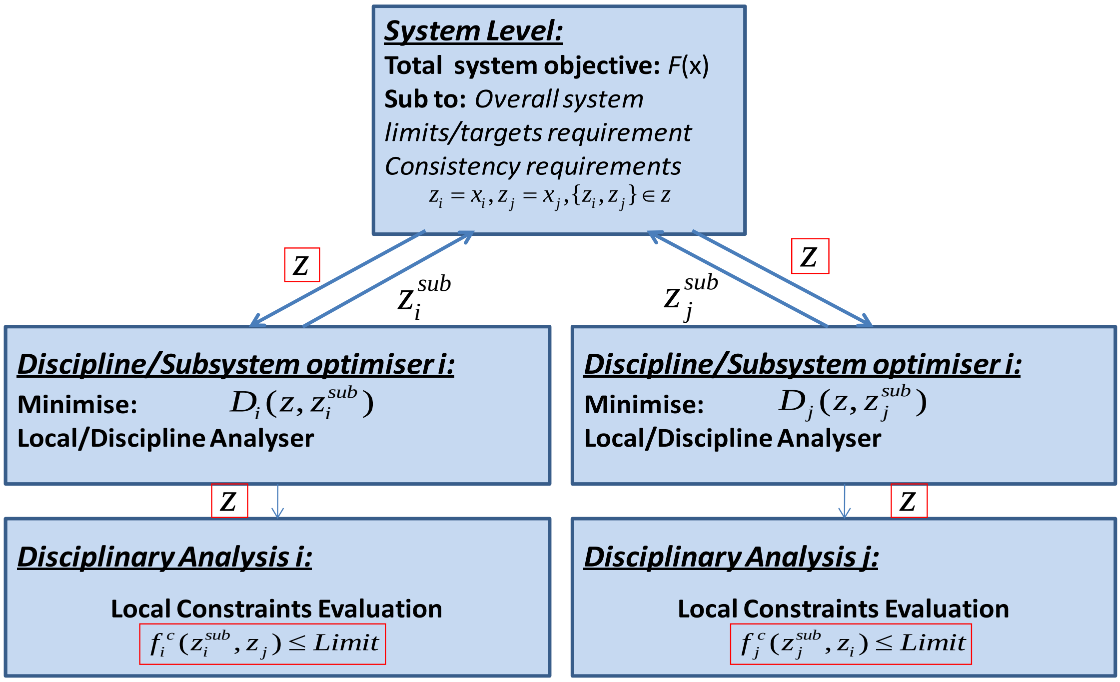


Figure 6.2 Demonstration of decomposing the strong coupling into discipline level

With strong and high dimensionality coupling, the decomposition may cost an excessive computation penalty. But for a problem that only requires a few auxiliary variables, the simplicity of the decomposition makes it worthwhile (Kroo, 2004).

6.2.1 TCO Formulation for a Diesel Engine Calibration

In analysing the steady-state calibration process to formulate a TCO framework, it was clear that the hierarchical structure already exists in terms of the “Global level / Drive Cycle“ and “Local level / minimap points“, as illustrated in Figure 6.3. The coupling between disciplines (as illustrated in Figure 6.3) is in terms of the maximum actuator change between the minimap points, which from a practical point of view could be associated with ‘driveability’ as one of the customer level satisfactions.

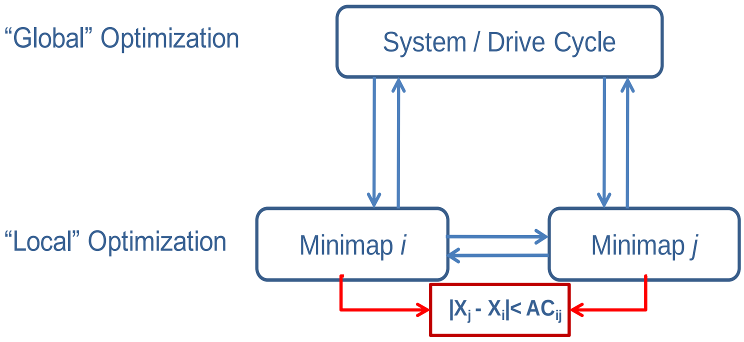


Figure 6.3 Multilevel structure of the engine calibration problem

In order to decompose this Diesel engine calibration problem, the disciplines/subsystems can be naturally defined as the 'local' calibration problem at each minimap point. By doing this decomposition, the constraints of smoothness actuator maps (Gradient Change constraints) can be removed from system level. Instead, the actuator GC constraints are performed individually at each discipline/subsystem (minimap point). This decomposition benefits the system level optimisation by releasing the GC constraints, which are recognised as the most difficult constraint from the constraints analyse in Chapter 5. From the literature review of MDO/CO (Kroo, 2004), subsystem optimisation was suggested for minimising discrepancy between local variables and system target value. Table 6.1 presents the analysis of the engineering objectives and targets for MDO/CO framework.

Table 6.1 MDO/CO- engineering analysis of objective and targets

	Discipline / Local level	System / Global level
Objectives		• *Minimise Fuel consumption over the drive cycle*
Targets	• Combustion stability (IMEP); • Engine Noise (Appolo); • Exhaust Temperature. • Drive ability – Smoothness of actuator maps in local area (around minimap point *i*) – gradient / actuator change constraint.	• Engineering limits for gaseous emissions (NOx, PM, HC, CO); • Consistency satisfaction with subsystem/disciplines feedback

Compared with the AAO analysis (Table 5.1), Table 6.1 shows that for the TCO framework the actuator map smoothness requirement (252 GC constraints) is removed from the system optimisation, which releases more freedom to the system optimisation. Instead, the Gradient Change constraints are handled by subsystem optimisation at corresponding disciplines (minimap points).

At system level, Equation 6.1 shows the optimisation problem formulation where total fuel consumption is minimised, which is defined as the system level objective ($J_{sys}(X)$), subject to the total emission constraints ($G(X)$).

The TCO solutions must also satisfy the consistency requirements among the disciplines by enforcing an equality constraint at the system level between system target and subsystem returned solution as a way to combine the interdisciplinary local optimal solution. Therefore, the system optimisation formulation is given, as follow:

Min: $J_{sys}(X) = \sum_{i=1}^{n} w_i * F_i(X) \qquad i \in (1, \dots n)$

Sub to:

Global constraints:

$$G(X) = \sum_{i=1}^{n} w_i * Emission_i < Emission\_Limit$$

Consistency constraints:

$$\sum (X_i^T - X_i^{sub})^2 = 0$$

Equation6.1

At the subsystem level, optimisation is carried out in parallel for all the minimap points. Equation 6.2 shows the formulation of the optimisation problem at each minimap point, which is to minimize the discrepancy between subsystem variable X_i^{sub} and system target value X_i^T, while satisfying local constraints.

Min: $J_{sub} = \sum (X_i^T - X_i^{sub})^2$

Sub to:

Local constraints:

$$H(X_i)_i = Local\ \_constraint_i < Local\_constraint\_limit_i$$

$$GC(X_i, X_j) = (X_i^{sub} - X_j^T) - GC\_limit_{ij} < 0 \qquad i \neq j, j \in (1,2, \dots node\ number)$$

Equation 6.2

This is also illustrated in Figure 6.4, that the steady state Diesel engine calibration problem is decomposed into a two level structure formulation of the TCO framework. Since the coupled constraints GC is decomposed, it needs to be performed in each discipline. Nevertheless, there is no local variables in one discipline for the others.

Therefore, the auxiliary variables are used as same as the variable from other disciplines. For example, where $X_i^{sub} - X_j^T$ is used same as $x_i - x_j$.

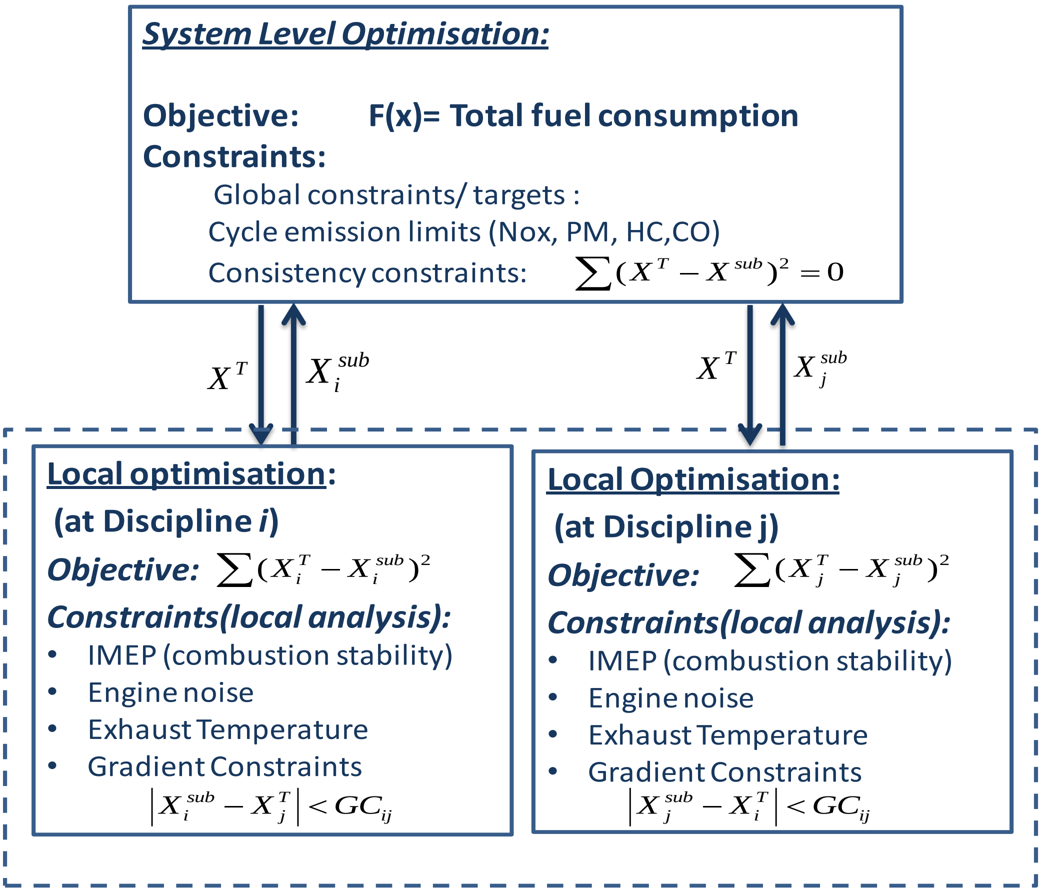


Figure 6.4 MDO/CO Formulation for Steady State Diesel Engine Calibration Problem

6.2.2 CO Implementation in MATLAB for a Diesel Engine Calibration

In order to manage the computational effort, different optimisation algorithms are employed at system and discipline level. At the system level, due to the large dimensional design problem, a GA algorithm was used to manage the optimisation task effectively. At each subsystem level optimisation, local variables are varied in order to find the local optimal solution, which is the closest feasible solution to the target values (X^T) passed from system level optimisation, as shown in Figure 6.4. As discussed in chapter 2, the gradient based searching algorithm can find the local optimal solution efficiently. Therefore, the gradient search algorithm *'fmincon'* (function in MATLAB) was employed at subsystem level optimisation. Similar to the AAO approach, a number of functions were created viz. the objective function and the system constraints function (shown in APPENDIX VI and VII).

The flowchart in Figure 6.5 illustrates the implementation of the TCO framework formulation with GA and Gradient based algorithms.

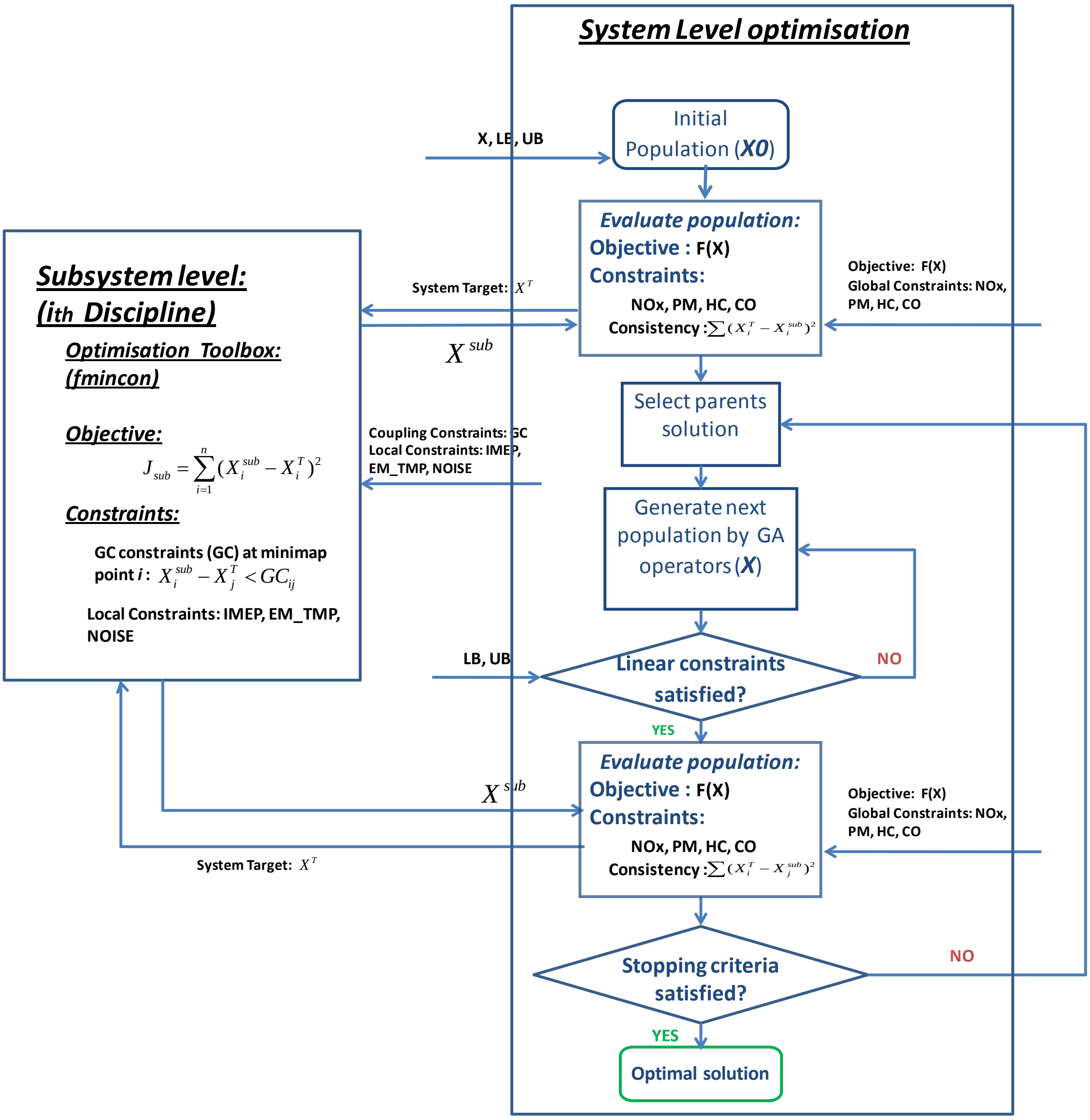


Figure 6.5 Illustration of TCO framework implementation for Diesel engine calibration optimisation

Figure 6.5 shows that at the system level, a Genetic Algorithm was used to minimise total fuel consumption subject to cycle emission constraints and enforced equality consistency constraints $\sum(x_i^T - x_i^{sub})^2 \leq 0$. With the GA standard algorithm, the initial population was generated and passed down to the subsystem level as the target value x^T. The subsystem optimiser aimed to minimise the discrepancy from the system target value with respect to GC constraints and local constraints. The discrepancy was formulated as $J_{sub} = \sum(x_i^T - x_i^{sub})^2$ and the GC constraints were formulated as $x_i^{sub} - x_j^T < GC_{ij}$ This optimisation process resulted in a local optimal solution that was the closest possible solution to the system target solution. The local optimal solution response was used to evaluate the enforced equality constraints afterwards. This procedure was carried out with all the population at every iteration of the GA process until the optimisation process converged. The developed Matlab scripts are shown in APPENDIX VIII.

6.3 TCO Results and Analysis:

To implement the TCO approach, the MATLAB based GA Algorithm was employed and the options were defined as indicated on Table 6.2, which is similar to the AAO implementation.

Table 6.2 Standard GA Option Settings

System Level		Subsystem Level	
Option name	**Setting**	**Options**	**Setting**
Selection Fcn	selectiontournament,2	**Display**	off
Popsize	100	**TolFun**	0.0001
Generations	200	**Algorithm**	active-set
Stall Gen Limit	4	**TolCon**	0.0001
Elite count	2% of population size	**TolX**	0.0001
Crossover fraction	0.6	**TolX**	0.0001
TolCon	0.01		
TolFun	0.01		
UseParallel	always		

6.3.1 Three minimap Points Results Analysis:

The test was also carried out with 3 nodes at first to test the feasibility of applying the TCO framework to the Diesel engine calibration problem. Figure 6.6 shows that the TCO approach delivered 3% improvement in terms of fuel consumption, compare to Two Step approach. However, this is worse than the AAO result.

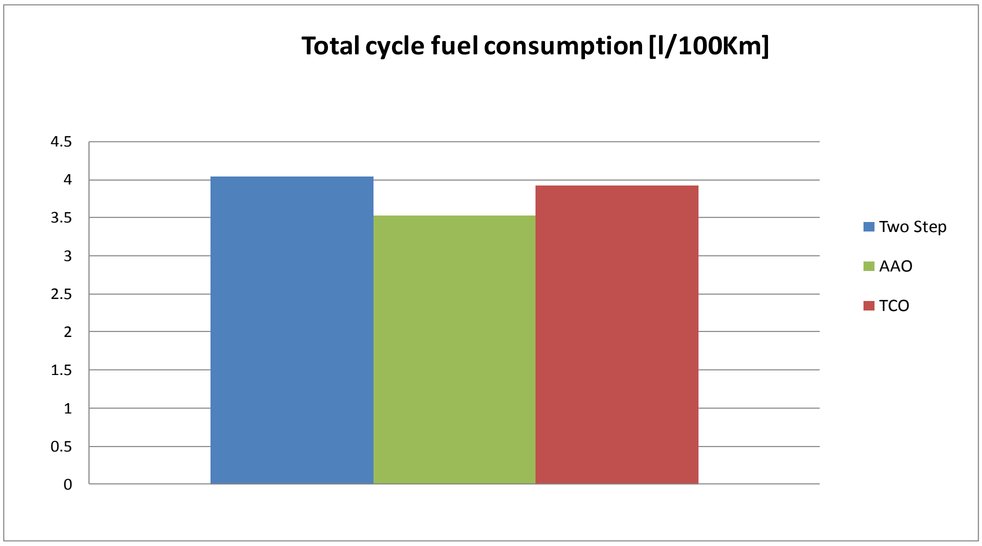


Figure 6.6 Comparison of fuel consumption for 3 nodes testing result

Figure 6.4 compares the results from 'Two Step', AAO and TCO approaches in terms of cycle emission results. These results are again represented according to NOx-Pm and HC-CO trade-offs with the engineering targets for total cycle emissions.

From the emission plots in Figure 6.7, it can be seen that all emissions of the optimal solution are within the engineering limits.

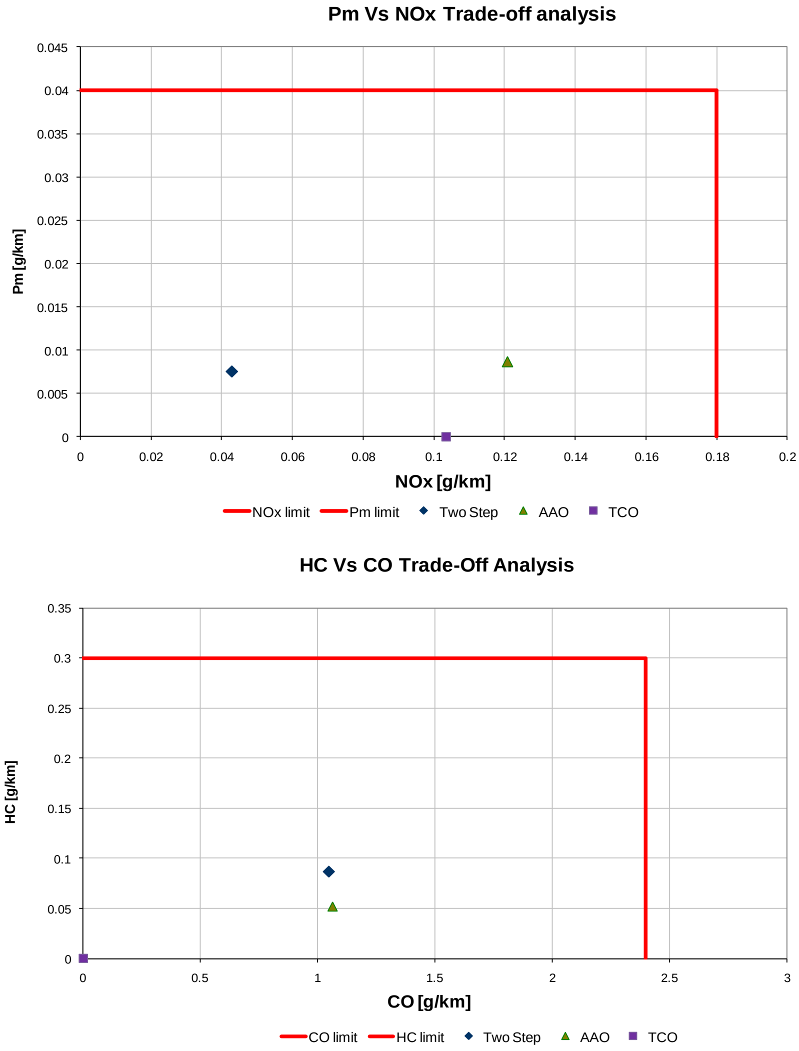


Figure 6.7 Comparison of Emissions from Different Approaches

In order to understand the effect of engine actuator settings, Figure 6.5 presents a summary of the optimal results in terms of actuator settings for 3 minimap points from the different approaches. The TCO optimal actuators settings plot in Figure 6.8 shows that the actuator settings of main injection timing, pilot injection timing and pilot injection quantity are distributed in quite different areas within the valid range compared to the “current” two-step process. Similar to AAO, th TCO optimal solution uses an increased pilot quantity to reduce the fuel consumption.

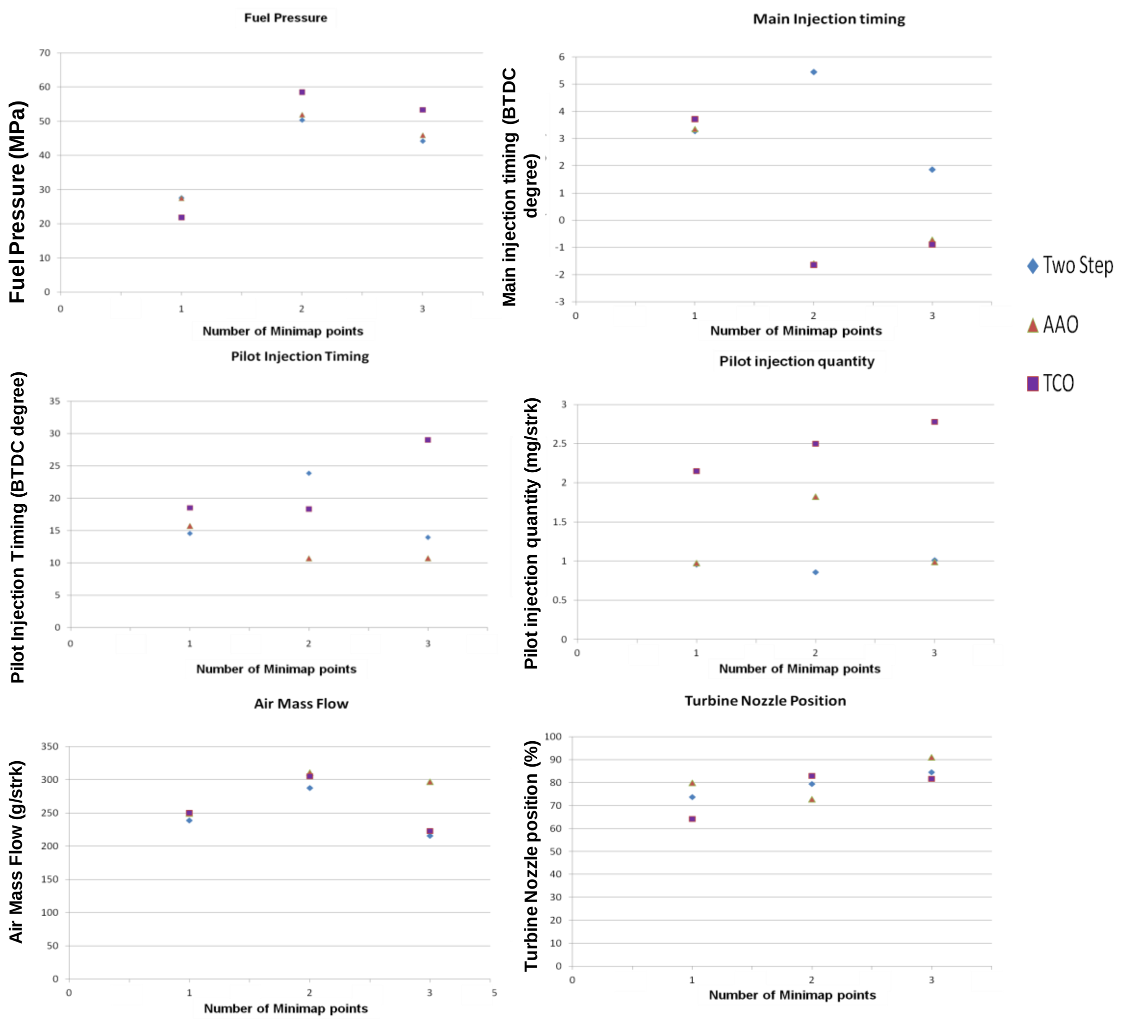


Figure 6.8 Illustration of optimal solution actuator settings from different approaches

6.3.2 Full Cycle Results Analysis:

As suggested by Kroo (Kroo, 2004), the TCO structure appears more beneficial for more complicated optimisation problems in terms of higher dimensionality, more constraints and strong coupling. Therefore, the TCO implementation for the whole drive cycle was carried out to assess the TCO framework formulation for Diesel engine calibration problem. Since the Diesel engine calibration optimisation still got a large amount of constraints and also to make the result comparable with AAO approach, the feasible population used in AAO were used within TCO approach as well.

Figure 6.9 shows the convergence throughout the optimisation process at the system level, the stopping criteria and the individual fitness value of the whole population. The convergence plot in Figure 6.9 shows that the objective value (fuel consumption) was reduced very sharply during the first 20 generations, then the reduction in the objective value flatted until about the 50th generation and the reduced rapidly again after the 50th generation. From the 60th generation the reduction was slowed down until it was fully converged. The obvious difference between the TCO convergence and typical GA convergence is discontinuous convergence of the TCO process, which is caused by the contribution of well exploring sub-design space at each discipline. This suggests the benefit of the TCO framework in addressing complicated engineering optimisation problems by better exploration of the design space.

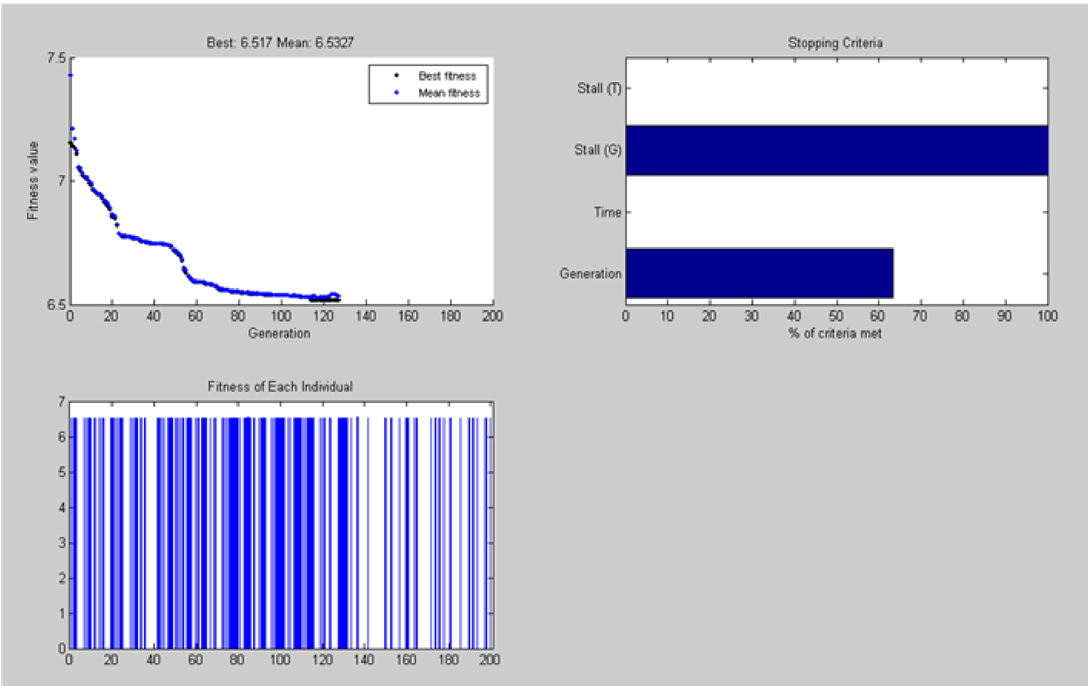


Figure 6.9 Illustration of TCO Optimisation Convergence with Stopping Criteria and Individual Fitness Values at System Level

Figure 6.10 illustrates a comparative analyse of optimisation results, showing that the fuel consumption was significantly improved (12%) obtained by the TCO approach. And it shows that the objective value (fuel consumption) of the optimal solution from Two-step approach was also slightly better than the AAO approach.

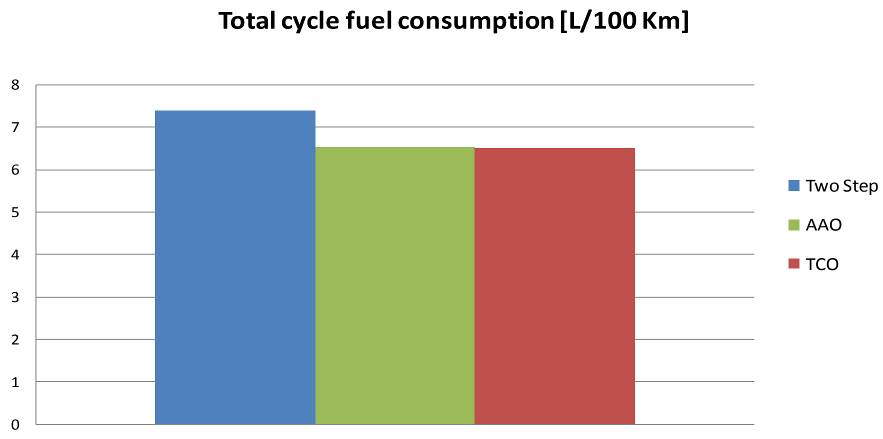
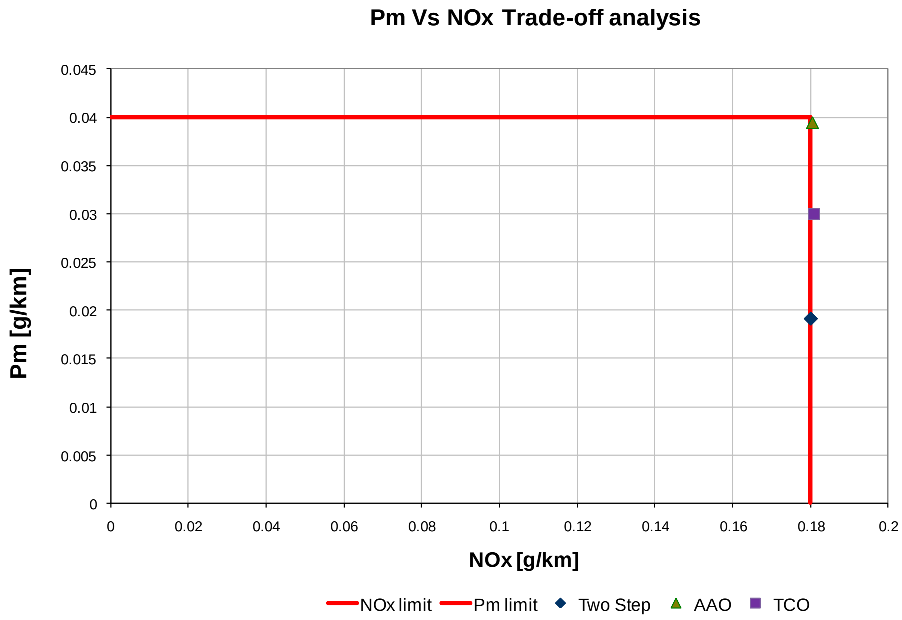


Figure 6.10 comparison of the optimal solution in terms of fuel consumption from different approaches

Figure 6.11 presents comparative results from different optimisation approaches in terms of total cycle emissions of NOx-PM and HC-CO with the engineering limits for the cycle emissions. From Figure 6.11, all the cycle emissions of the valid solutions that were generated by the TCO approach met the engineering cycle emission targets.



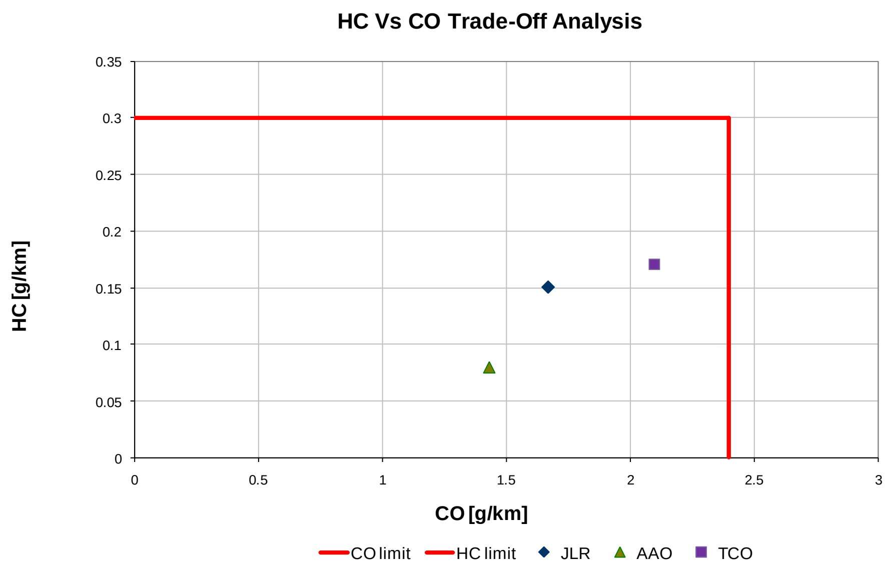


Figure 6.11 Results Analysis: cycle emissions

Figure 6.12 shows the summary of actuator settings from the valid solutions produced by the different optimisation approaches.

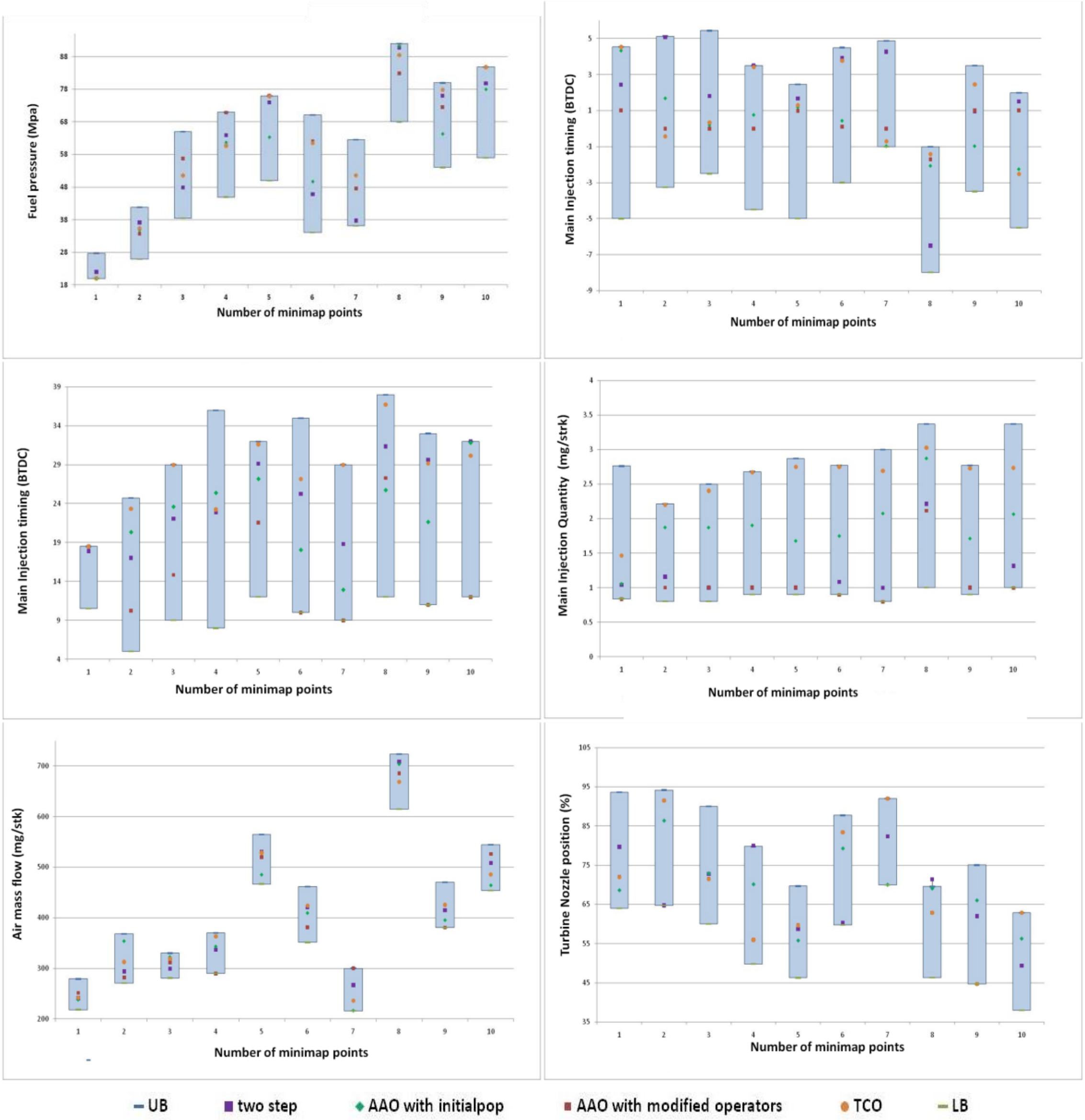


Figure 6.12 Results analysis: optimal actuator settings from different optimisation approaches

Figure 6.12 shows that the solutions from the different optimisation approaches are distributed in quite different areas within the valid range. Especially, optimal solutions for fuel pressure, pilot injection timing and pilot fuel quantity are quite different between the optimisation methods. From an engineering viewpoint, the TCO optimal solution uses a higher fuel pressure compared to the AAO solution, which has the effect of increasing temperature in the combustion chamber, which could improve combustion quality. The TCO solution also suggests retarded pilot injection timing and less pilot fuel injected. This difference between AAO and TCO optimal solutions with 0.02L/100Km improvement of fuel economy just proved that the TCO optimisation approaches can provide better exploration of the design space during the searching process.

Table 6.3 summarizes comparatively the results from different optimisation methods in terms of the main objective (fuel consumption) and computation expense (Seabrook). It is seen that while the TCO and the AAO approaches have comparable fuel economy benefit, the computing time for the TCO approach is considerably better than AAO, which can be attributed to the restructuring the optimisation problem. This clearly shows the TCO framework has a strong ability to deal with complicated engineering system optimisation problems.

Table 6. 3 Summarise of fuel consumption from different optimisation approaches

	Fuel Consumption (L/100Km)	Comparison	Computing time (Hrs)
Two-step	7.38		
AAO	6.53	↓ 12%	62.8
TCO	6.52	↓ 12%	41.52

Table 6.4 shows comparatively the cycle emissions from the Two Step, AAO and TCO approaches. The AAO solution has kept the NOx emission level and decreased CO and HC emissions, but increased the PM emissions by 107% (still within the emissions limit). The TCO solution has kept the NOX emissions about same level and increased all the other cycle emissions but still well inside the engineering target.

It can be concluded that both the AAO and TCO approach have produced better fuel economy while meeting the emissions targets, with TCO delivering a computation advantage over AAO

Table 6.4 Comparison of Emissions from Different Approaches

Cycle Output	*Two-step*	*AAO*	*AAO Vs Two-step*	*TCO*	*TCO Vs Two-step*	*Legislation Limits*
cycle_NOx_g/km	0.1800	0.1804	⇒ 0%	0.1809	⇒ 0%	0.18
cycle_CO_g/km	1.6697	1.4310	↓ 14%	2.0980	↑ -26%	2.4
cycle_HC_g/km	0.1505	0.0800	↓ 47%	0.1707	↑ -13%	0.3
cycle_PM_FSN_g/km	0.0191	0.0395	↑ -107%	0.0300	↑ -57%	0.04

6.5 Discussions and Conclusions:

The Diesel engine calibration optimisation problem was formulated as a hierarchical multidisciplinary optimisation under a two level hierarchical TCO framework. A penalty function was employed to ensure system consistency. The results show that a better solution was produced in terms of fuel consumption; compared with the two-step approach, fuel economy was improved by 12%, and emissions were just within tolerance of the emission limits.

Compared with the two-step approach, computation time was improved by decomposing the large scale optimisation problem into a number of smaller subsystems, as shown in Table 6.4. The TCO framework has the advantage that it can be easily expanded to include more variables and disciplines/subsystems.

The TCO implementation of the Diesel engine calibration optimisation problem still has some disadvantages, such as:

- To decompose the strong coupling between disciplines (minimap points), the Gradient Constraints have been performed at each discipline (minimap point), which means the decomposition just doubles the gradient constraints. Although, the local/discipline optimisations have been carried out in parallel, it still costs expensive computation effort.
- Since the consistency constraints are hard to achieve for the system optimisation, it was seen that this affected the ability to find the feasible solution for the initial population. In order to address this issue the "Creator" operator has been used to generate a feasible initial population,
- Similar to AAO, this framework requires a significant of prior calibration knowledge such as the maximum actuator changes; such information might not be easily available for new engines and applications, and in any case it is a very time consuming task.

Chapter 7 Further Development of Collaborative Optimisation for Diesel Engine Calibration

7.1 Introduction

The previous 2 chapters presented an analysis of the Diesel calibration problem within a Multidisciplinary Design Optimisation framework, and the implementation of two conventional MDO frameworks (All-At-Once and Collaborative Optimisation). The results from both AAO and TCO implementations showed a considerable improvement in terms of quality of optimal calibration solutions over the conventional two-step process presented in Chapter 4.

It has been shown and discussed that the main challenge of Diesel calibration optimisation is that both the objective and the solution spaces are heavily constrained. While the AAO and TCO implementations presented have successfully dealt with this issue, this success in terms of delivering a good solution to the optimisation problem comes with a severe penalty in terms of computational expense.

The aim of this chapter is to present the development and implementation of a refined Collaborative Optimisation based framework, which attempts to better deal with the complexity of Diesel calibration optimisation.

7.2 Diesel Calibration Problem: Review and Analysis

7.2.1 Review and Analysis of Problem Difficulty

In the Diesel engine calibration optimisation problem, the computational effort lies mostly with the constraints analysis. There are several different kinds of constraints imposed, reflecting both legislative requirements for emissions and calibrator's preferences for engineering attributes (such as noise and driveability). From a calibration optimisation point of view, constraints are imposed both in the solution space (mainly reflecting calibrator's preferences for strategy (i.e. domain constraints for actuators) and smoothness of the actuator map) and in the objective domain (associated with emissions performance and other attributes such as noise, combustion stability, etc).

The linear constraints define each engine actuator's valid range. Cycle emission constraints are the legal limits of emissions over the whole drive cycle calculated as the weighted sum of the emissions at each minimap point. GC constraints define the actuator maximum allowable change between different minimap points. Local constraints define the limits of acceptable engine performance produced at each engine condition. As the calibration optimisation problem analyse in section 4.5, cycle emission limits are required in order to ensure satisfaction of emission legislations. Similarly, local engine constraints must also necessarily be maintained.

The constraints analysis presented in Chapter 5 has shown that the most difficult constraints to satisfy are cycle emission constraints (especially NOx/PM) and Gradient Constraints (GC). There are 252 actuator gradient change constraints between the minimap points. Within the AAO framework formulation, the actuator map smoothness was addressed by the integration of GC constraints between the minimap points and was formulated as a global constraint. This resulted in considerable difficulty with finding feasible solutions, which leads to either very expensive computation or an unreliable optimisation process. The TCO framework strategy was to simplify the system optimisation problem by breaking down these couplings (GC constraints) to each discipline (minimap point). Since each of the GC constraints involves two minimap points, the GC constraints analysis is required to be performed in both of those disciplines. Therefore, the computational effort for the GC constraints analysis is doubled. However, this decomposition has relaxed the freedom for the system optimisation, and ultimately this has been shown to deliver an improvement in computational speed compared to AAO.

7.2.2 Optimisation Problem Re-formulation

The conventional wisdom on difficult constrained optimisation problems Forrester (Forrester et al., 2008) suggests that removing constraints is usually an effective way of tackling the problem. This principle has been already demonstrated through the TCO framework which has effectively removed the GC constraints from the system level optimisation, delegating these to subsystems. The fundamental idea for further development of the Diesel calibration optimisation problem is based on removing the GC constraints altogether, which would further simplify the computational problem.

This approach can be justified from the engineering analysis of the calibration problem. The calibration preference is for a "smooth actuator map". This preference could be formulated mathematically in many different ways; calibrators at the Sponsoring Company have formulated this as a "maximum actuator change constraint" between minimap points (referred to as a GC constraint in this thesis).

Other approaches seen in the literature (Pienaar, 2007 , Roudenko et al., 2002) have formulated this directly in terms of a “gradient” constraint, defined as maximum change per unit increase in engine speed / load. Implementing such an approach would require extensive analysis of what is an acceptable gradient for each actuator, and in terms of optimisation implementation would still require a large number of evaluations against the limiting value.

Reflecting upon the fact that the ideal engineering calibration requirement is for the actuator map to be as smooth as possible, pointed to the notion that an ideal formulation of this preference would in fact be to minimise the gradient, or the actuator change, between minimap points. The implication for the Collaborative Optimisation implementation would be that in order to address the requirement for a smooth actuator map, the subsystem optimisation problem should be redefined from a ”discipline feasible solution analysis” to a ”minimisation of actuator gradient / change”.

With this approach the idea is that subsystem level optimisation would be carried out to produce an engine control map that was as smooth as possible in every local area (load/speed space) presented in each discipline, as well as ensuring consistency with the system level optimisation.

Figure 7.1 illustrates the structure of this revised Collaborative Optimisation framework, termed Hybrid Collaborative Optimisation (HCO). In a similar manner to TCO, at system level the total fuel consumption is minimised subject to cycle emissions constraints and the consistency constraints of subsystem responses. The system target value is passed down to each discipline, and at each discipline the “local smoothness” (i.e. actuator change)is minimised subject to the local constraints.

In order to ensure compatibility with the system level optimisation a constraint is imposed on the system overall objective, i.e. the subsystem task is to deliver the minimal actuator changes with the 'not worse' fuel economy as the system target, which is shown in Figure 7.1.

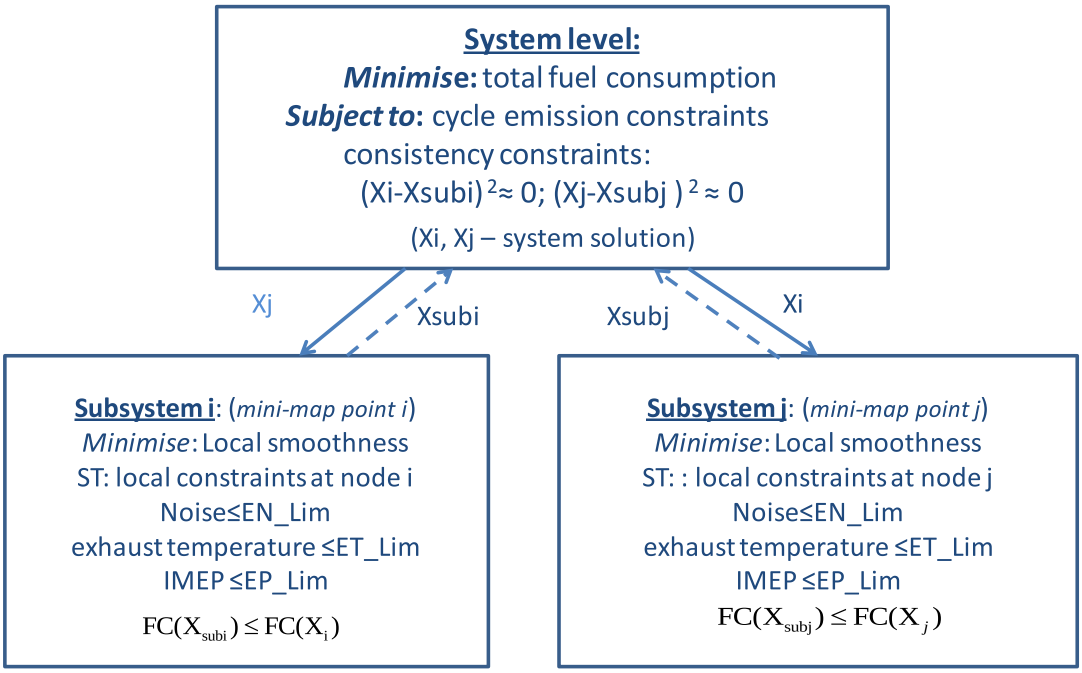


Figure 7.1 HCO data flow structure

7.3 Formulation and Implementation

7.3.1 Mathematical Formulation of "Smoothness"

As suggested in Section 7.2, the engine actuator control map is optimised at discipline (minimap) level by using subsystem optimisation rather than by using actuator gradient change constraints. To carry out the local/discipline optimisation, the engine actuator map smoothness objective needs to be formulated at each minimap point.

As illustrated in Figure 4.11, an immediate formulation of smoothness at a local discipline (minimap) *i* can be expressed as the sum of the absolute values of the actuator changes associated with all possible transitions from node *i*, which is formulated in Equation 7.1.

Smoothness1:

$$J_{smooth}(X) = \sum_{j=1}^{n} \left|(X_i - X_j)\right|$$

Equation 7.1

X_i is the actuator settings at minimap point i and X_j is the actuator setting at minimap j with the possible transitions from minimap point i, as illustrated in Figure 4.11. The issue with this smoothness formulation is that it involves a number of actuator changes that are expressed in different engineering units (scales). For example, the scale of mass air flow and injection timing are very different. Since the smoothness formulation in Equation 7.1 is taking the sum of actuator change as equal weight, the optimisation algorithm would prioritise the mass air flow rather than the injection timing as it would likely lead to higher improvement from generation to generation. Therefore, this smoothness formulation is not going to be involved any more, as more effective smoothness formulation is going to be developed.

The GC constraint was defined in Chapter 4 as the maximum actuator allowable change, which is expressed in the engineering unit associated with each actuator. The local actuator map smoothness can be normalised across the different actuators by taking the ratio of the actual actuator change between two minimap points to the maximum allowable actuator change. Based on this idea, an overall objective function for the discipline can be derived as the sum of squared normalised actuator smoothness at the particular minimap point, as shown in Equation 7.2.

Smoothness2:

$$J_{smooth} = \sum_{j=1}^{n} \left(\frac{(X_i - X_j)}{GC_{limit_{ij}}} \right)^2$$

Equation 7.2

The subsystem optimisation with the smoothness in Equation 7.2 is not only standardising the actuator change between the actuator control variables, but also standardising gradient change across the map. Specifically, the allowed gradient should be smaller on load increases, which cause drive-ability issues, than on speed increases. The valid range for different minimap points has taken this into account, therefore by dividing to the allowable actuator change (carefully defined by calibrators) you account for these criteria as well.

An alternative approach to formulating the local objective function is based on a mini-max principle, i.e. minimise the maximum normalised actuator change around a minimap point. The mathematical formulation for this approach is shown in Equation 7.3.

$\boldsymbol{smoothness3}$:

$$J_{smooth}(X) = \max \left(\frac{(X_i - X_j)}{GC_{limit_{ij}}} \right)^2$$

Equation 7.3

7.3.2 HCO Mathematical Formulation

As discussed, the HCO formulation follows the Collaborative Optimisation framework structure. Thus, the system formulation of HCO is similar to the TCO system optimisation formulation.

The optimisation problem formulation at system level is shown in Equation 7.4, where total fuel consumption is minimised as system level objective ($J_{sys}(X)$).

System level:

Minimise: $J_{sys}(X) = \sum_{i=1}^{n} w_i * F_i(X) \qquad i \in (1,2, \ldots node\ number)$

Subject to:

Global constraints:

$$G(X_i) = \sum_{i=1}^{n} w_i * Emission_i < Emission\_Limit$$

Consistency constraints:

$$H(X_i) = \sum_{i=1}^{n} (X_i - Xsub_i)^2 \qquad \textbf{\textit{Equation 7.4}}$$

w_i is the residency time of each minimap point contributing to the NEDC cycle; F_i is the fuel consumption for minimap point *i*; and *X* is the vector of control variables (*G(X)* denotes the total emission constraints). The solutions must also satisfy the consistency constraints ($H(X_i)$) among the disciplines, expressed as equality constraints at the system level that coordinate the interdisciplinary couplings.

At subsystem level, the optimisation problem can be described as Equation 7.5.

Subsystem $\boldsymbol{i}$:

Minimise: $J_{smooth}(X_i^{sub})$

Subject to: *Local constraints:*

$$NOISE(X_i^{sub}) < Local\_NOISE\_limit_i$$

$$Exhaust\ TEMP\ \left(X_i^{sub}\right) < \ Local\_ExhaustTEMP\_limit_i$$

$$IMEP\left(X_i^{sub}\right) < \ Local\_IMEP\_limit_i$$

$$FC\left(X_i^{sub}\right) \leq FC(X_i^T)$$

Equation 7.5

Since the local smoothness was used as the objective at subsystem level, the GC constraints have been removed. Consequently, the local constraints are used to satisfy the engine performance requirements and the fuel consumption constraint is used to ensure not worse than system target. The J_{smooth} is the subsystem objective, as discussed in last section, X_i^{sub} is the vector of local variables at minimap point $i$and the X^T is the vector of target variables passed from system level, which are same as the system variables X. Therefore, the smoothness formulation can be written as:

$$J1_{smooth}(X) = \sum_{j=1}^{n} \left(\frac{\left(X_i^{sub} - X_j^T\right)}{GC_{limit_{ij}}} \right)^2$$

$$J2_{smooth}(X) = \max \left(\frac{\left(X_i^{sub} - X_j^T\right)}{GC_{limit_{ij}}} \right)^2$$

Equation7.6

Both $J1_{smooth}$ and $J2_{smooth}$ been implemented in the HCO framework, algorithms referred to as "HCO V1" (corresponding to $J1_{smooth}$) and "HCO V2", respectively (corresponding to $J2_{smooth}$).

7.3.3 MATLAB Implementation

The MATLAB implementation has been developed based on the Collaborative Optimisation structure. New scripts had to be developed only for the subsystem optimisation, shown in *APPENDICES* X to XII. Figure 7.2 illustrates the implementation of the HCO development based on TCO.

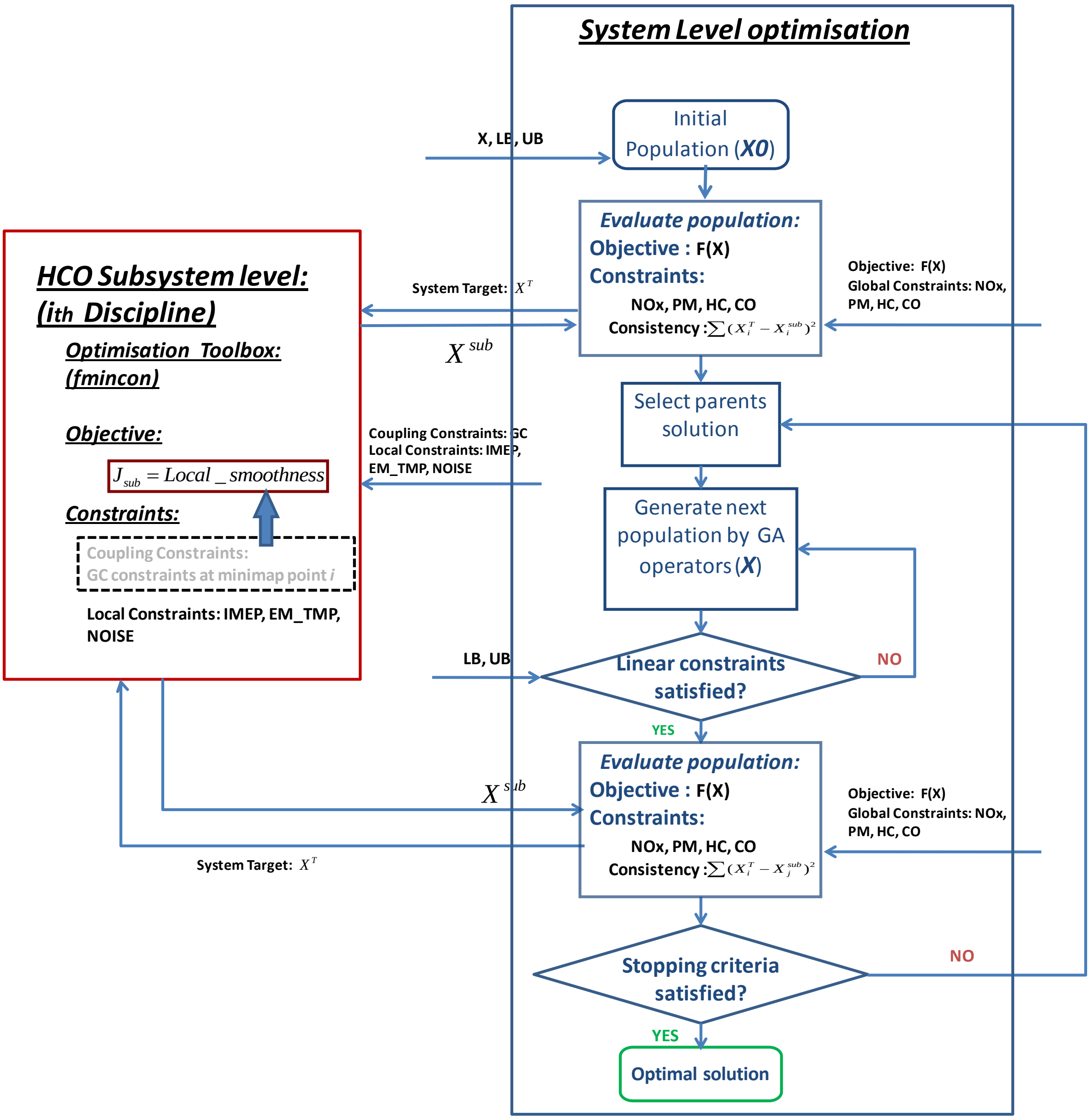


Figure 7.2 Illustration of Development of HCO based on the TCO Framework with Integrated Subsystem Optimiser

Table 7.1 shows the options for both system and subsystem level optimisation options for the standard MATLAB algorithm.

Table 7.1 Optimisation Options Setting for System Level and Subsystem Level

System level		Subsystem level	
Options	**Setting**	**Options**	**Setting**
Selection Fcn	selectiontournament,2	**Display**	off
Popsize	100	**TolFun**	0.0001
Generations	200	**Algorithm**	active-set
Stall Gen Limit	4	**TolCon**	0.0001
Elite count	2% of population size	**TolX**	0.0001
Crossover fraction	0.6		
TolCon	0.01		
TolFun	0.01		
UseParallel	always		

7.4 Tests and Results

7.4.1 Three minimap point test

As with the implementation of the previous optimisation frameworks (AAO and TCO), a test for a reduced data set (three minimap points) was carried out first to explore the capability of the formulation to deal with the Diesel engine calibration problem.

Figures 7.3 and 7.4 show the convergence plots throughout the optimisation process at system level, the stopping criteria, individual fitness values of the whole population, and the smoothness plot for three different formulations of local smoothness. In order to compare the smoothness plot between different formulations, the smoothness was calculated as the maximum ratio between actuator change and corresponding GC constraint on each actuator, which is same as the local smoothness formulation in Equation 7.3. Figures 7.3 and 7.4 show the convergence of HCO approaches with the different subsystem formulations. To compare these optimisation plots, The HCO with the subsystem formulation of $J2_{smooth}$ has only taken less than 5 iterations to achieve the objective function value of 4.5, which the HCO with the subsystem formulation of $J1_{smooth}$ has taken about 70 iteration to achieve. However, the HCO optimisation test with $J1_{smooth}$ formulation used the population size of 50 and the HCO optimisation test with $J2_{smooth}$ formulation used the population size of 100. The HCO optimisation test with $J2_{smooth}$ formulation has shown a great opportunity to find a better solution in terms of minimising fuel consumption compare with formulation of $J1_{smooth}$.

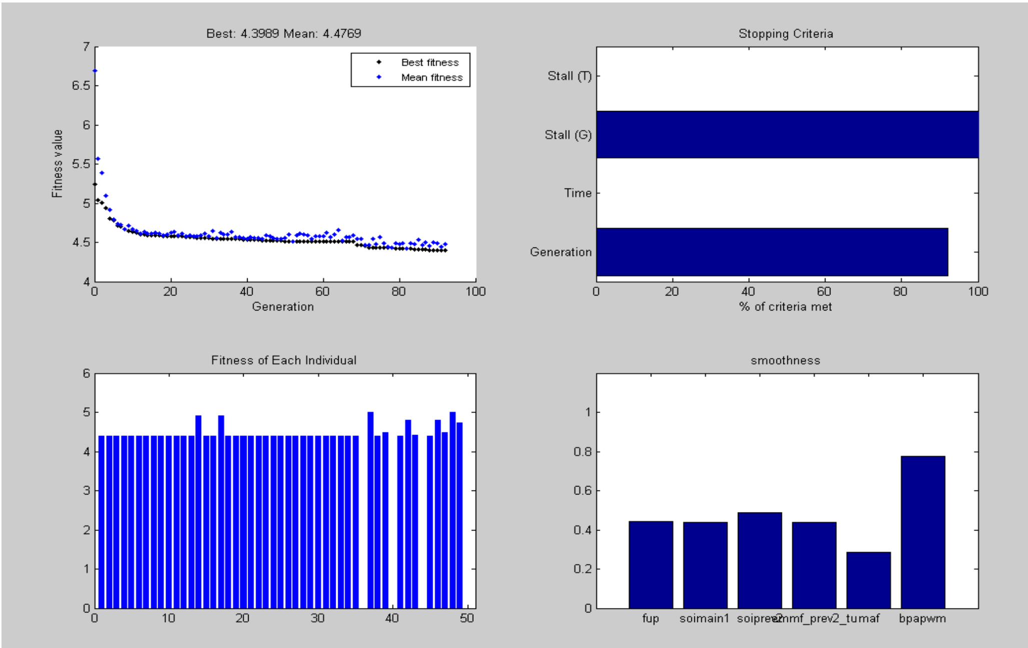


Figure 7.3 Illustration of HCO Optimisation Convergence with the formulation $J1_{smooth}$

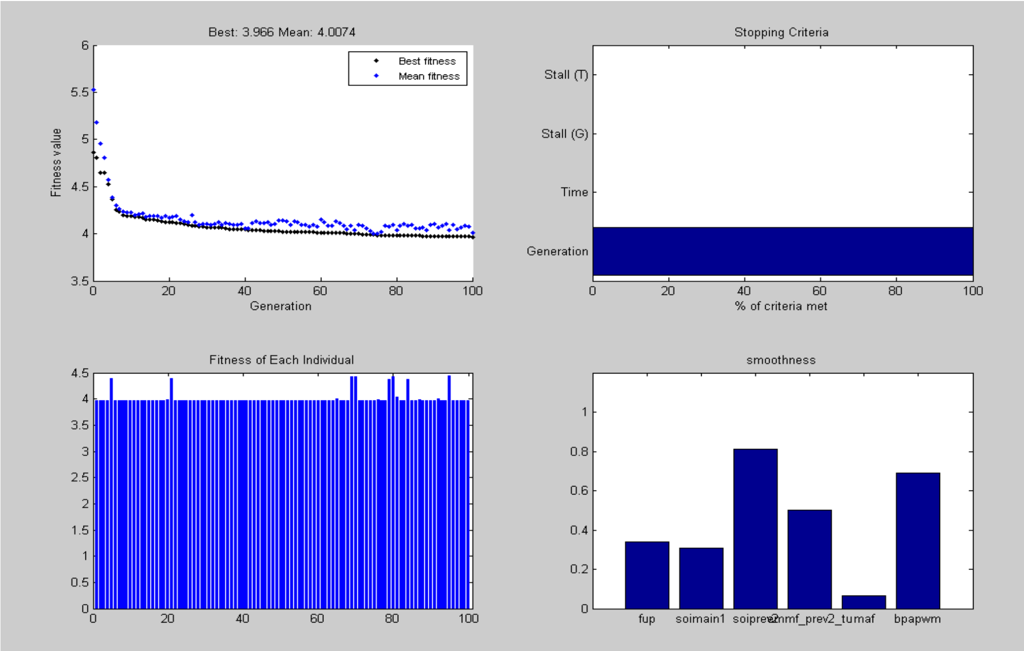


Figure 7.4 Illustration of HCO Optimisation Convergence with the formulation $J2_{smooth}$

Figure 7.5 compares the fuel consumption between the optimal solutions for the different smoothness formulations.

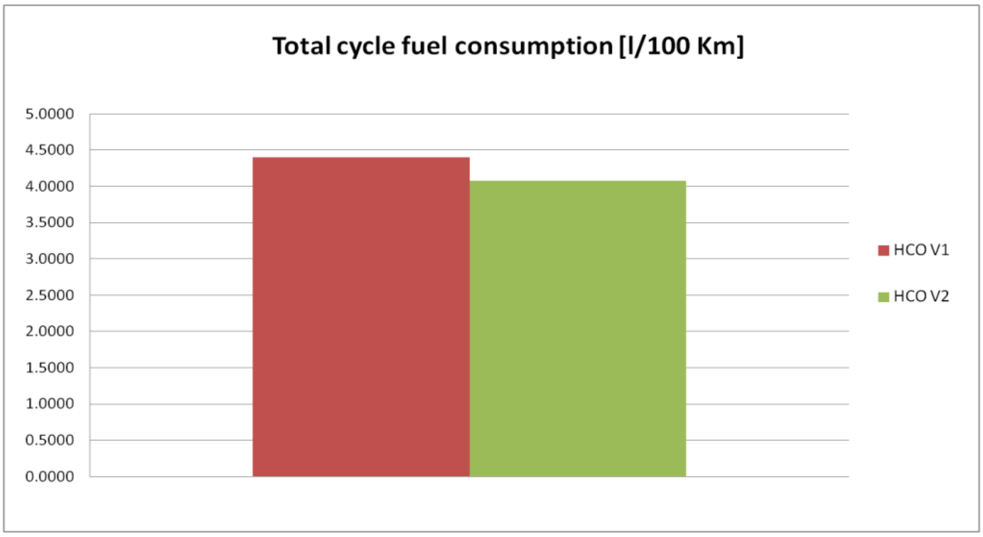


Figure 7. 5 The comparison of the optimal solution of fuel consumptions between HCO V! and HCO V2

Figure 7.6 shows that both HCO approaches produced acceptable results from the emissions point of view.

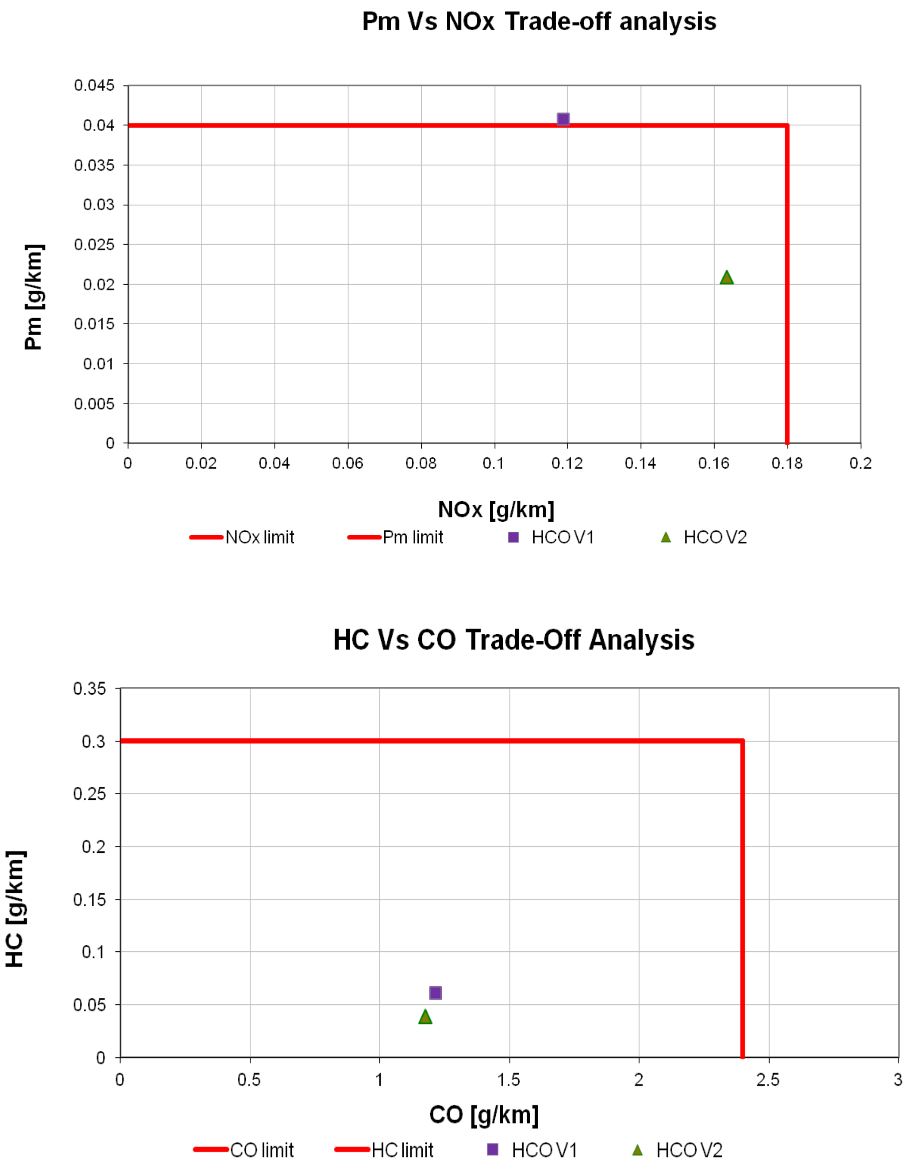
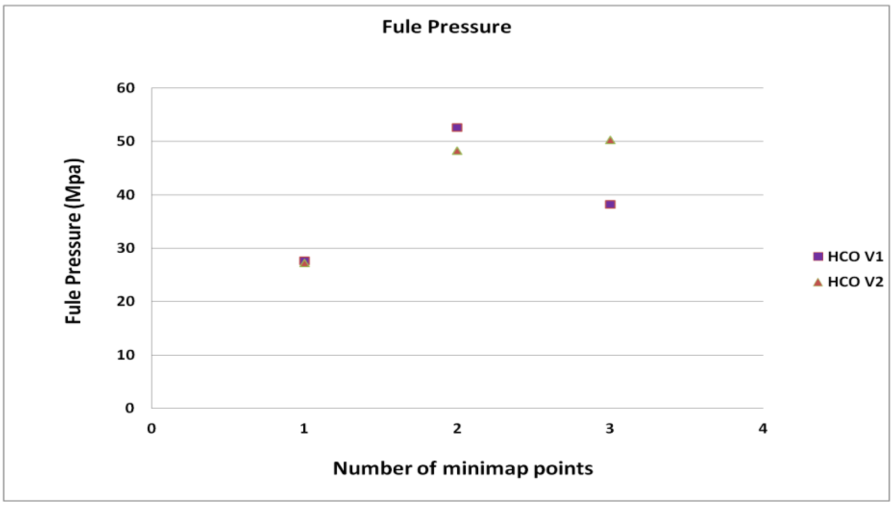
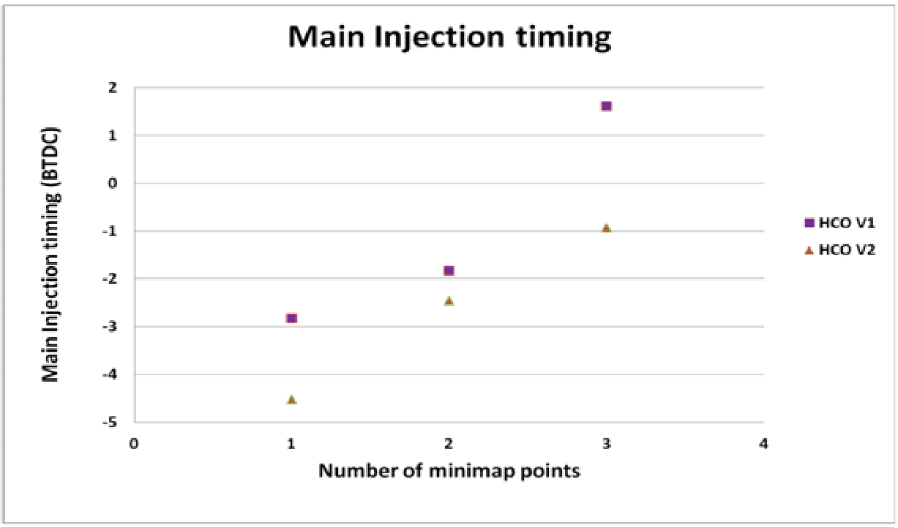
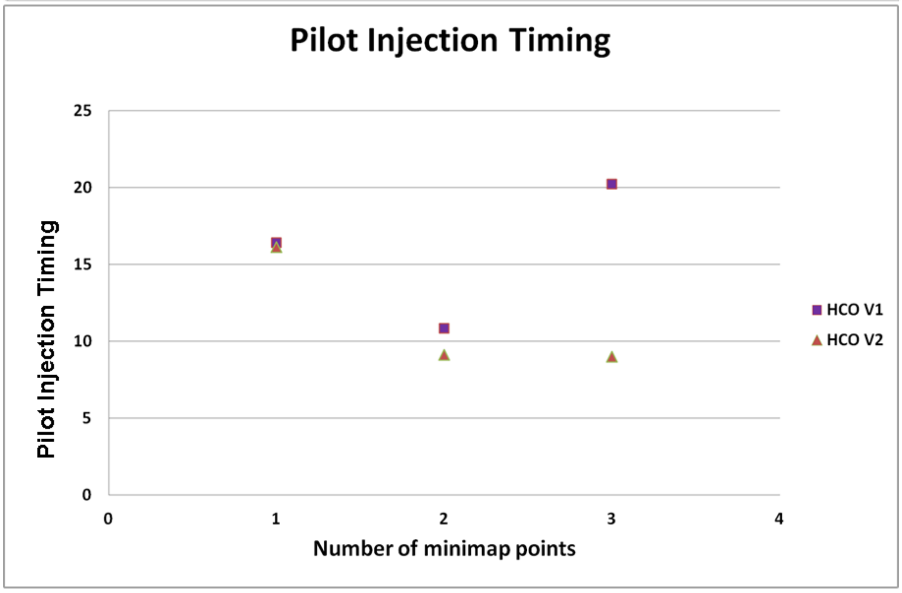


Figure 7.6 HC Vs CO Emissions plot for different HCO V1 and HCO V2

Figure 7.7 illustrates the actuator settings solutions from different HCO formulations. The two set of solutions are obviously follow the different trend, which the solution produced by the framework with the $J2_{smooth}$ formulation are smoother than the $J1_{smooth}$ formulation in all of the actuator settings.


Fule Pressure
Fule Pressure (Mpa)
60
50
40
30
20
10
0
0
1
2
3
4
Number of minimap points
HCO V1
HCO V2


Main Injection timing
Main Injection timing (BTDC)
2
1
0
-1
-2
-3
-4
-5
0
1
2
3
4
Number of minimap points
HCO V1
HCO V2


Pilot Injection Timing
Pilot Injection Timing
25
20
15
10
5
0
0
1
2
3
4
Number of minimap points
HCO V1
HCO V2

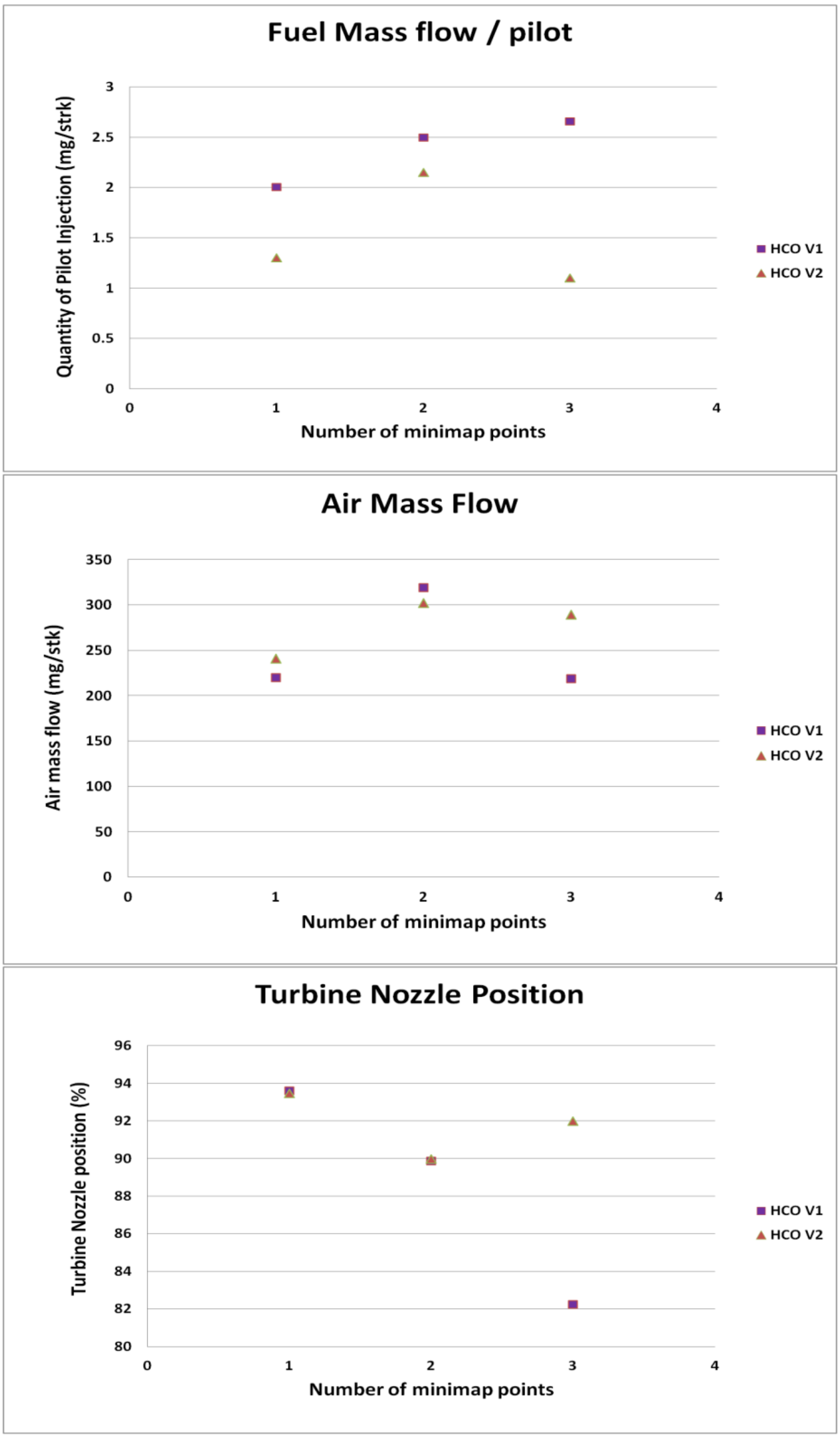


Figure 7.7 The optimal solution actuator settings from different HCO approaches

In order to compare the HCO with the other approaches, the best optimal solution was chosen in terms of fuel consumption. Figure 7.8 illustrates a comparison of the optimal solutions for 3 minimap points from different approaches: two-step, AAO, TCO, and HCO. The comparison of fuel consumption clearly indicates that AAO was the best approach, improving fuel consumption by about 15% as compared to the two-step, TCO, and HCO approaches.

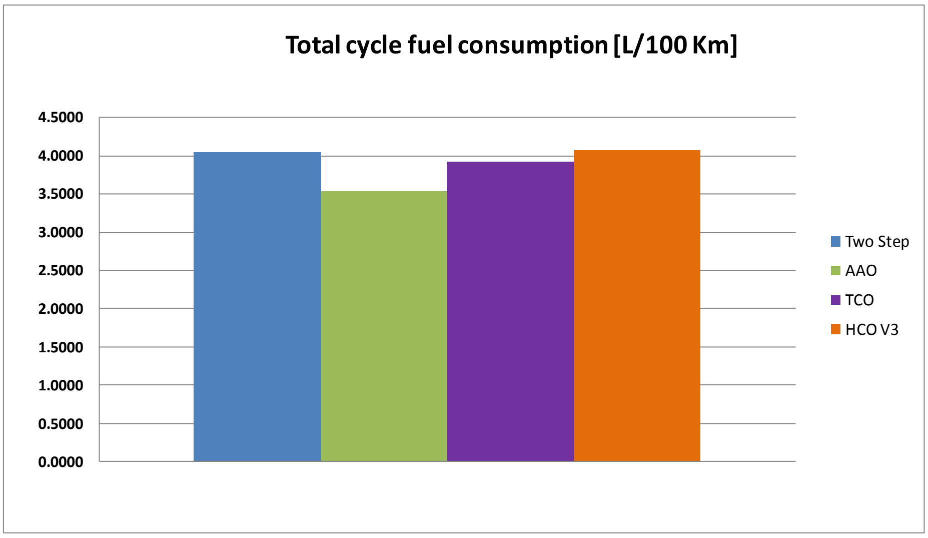


Figure 7.8 Comparison of fuel consumption between two-step, AAO, TCO, and HCO

7.4.2 Tests and results for 10 minimap points

In the MDO literature (Kroo, 2004 , Sobieszczanski-Sobieski and Haftka, 1997 , Sobieszczanski-Sobieski and Kroo, 1996) it is discussed that Collaborative Optimisation based frameworks should perform better with optimisation problems having more dimensionality and constraints, which suggests that it is worth testing the HCO approach with the full cycle calibration optimisation problem.

Since more subsystem optimisations are involved, the consistency constraints are harder to satisfy. In order to maintain the solution in the population while forcing the search direction to the feasible area, a simple penalty function has been employed, which add the discrepancy between the subsystem and system solution into the objective function with same scale, as shown in Equation 7.7.

System Level:

Minimise: $J_{sys}(X) = \sum_{i=1}^{n} w_i * F_i(X) + \ Coef_{pen} * (\sum_{i=1}^{n}(X_i - Xsub_i)^2)$

Subject to: $G(X_i) = \sum_{i=1}^{n} w_i * Emission_i < Emission\_Limit$

Equation 7.7

For the implementation, in order to make the results comparable with the AAO and HCO results, the population size was chosen to be 200. Figure 7.9 illustrates the convergence throughout the optimisation process at system level, the stopping criteria, and individual fitness values of the whole population for 10 minimap points (with smoothness plots).

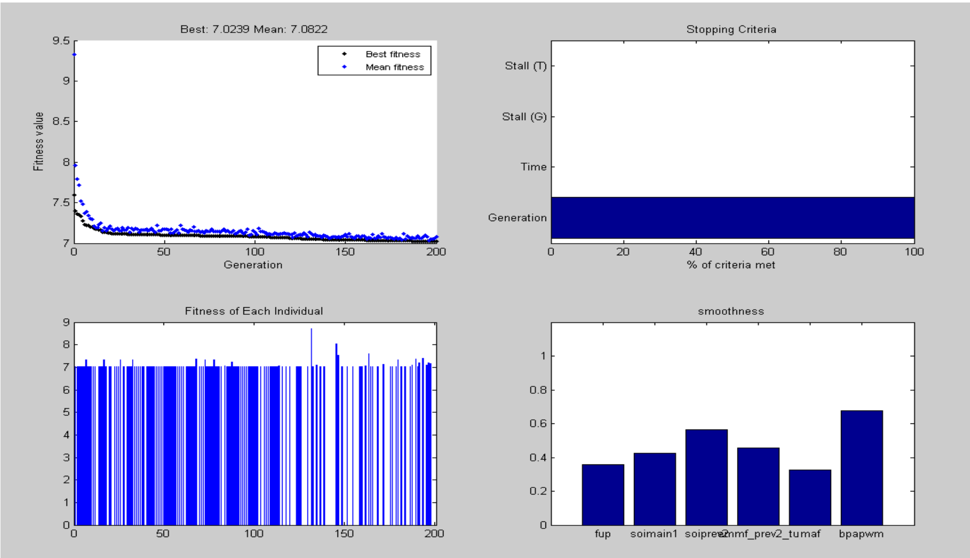


Figure 7.9 Illustration of HCO Optimisation Convergence with Stopping Criteria and Individual Fitness Values at System Level with smoothness plot

Figure 7.10 shows the result for fuel consumption for the HCO approach with subsystem formulation of $J2_{smooth}$, which was observed to produce the best results during several trials. This results shows that HCO improved the fuel economy by 5.5% (reduction of fuel consumption). The relatively lower performance in terms of fuel economy compared to AAO and TCO is due to the fact that the optimisation objective included a penalty function on the consistency constraints, rather than just minimising fuel consumption as in AAO and TCO.

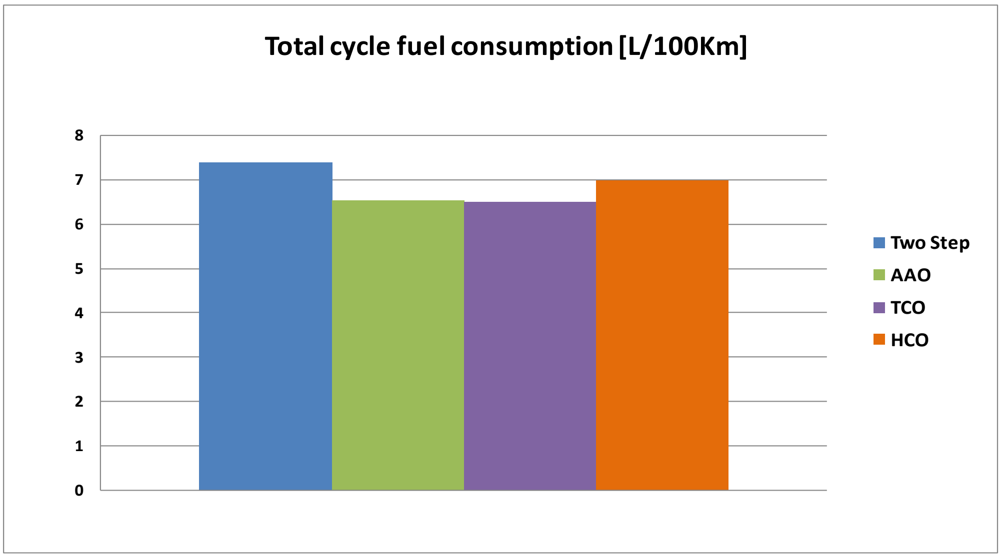
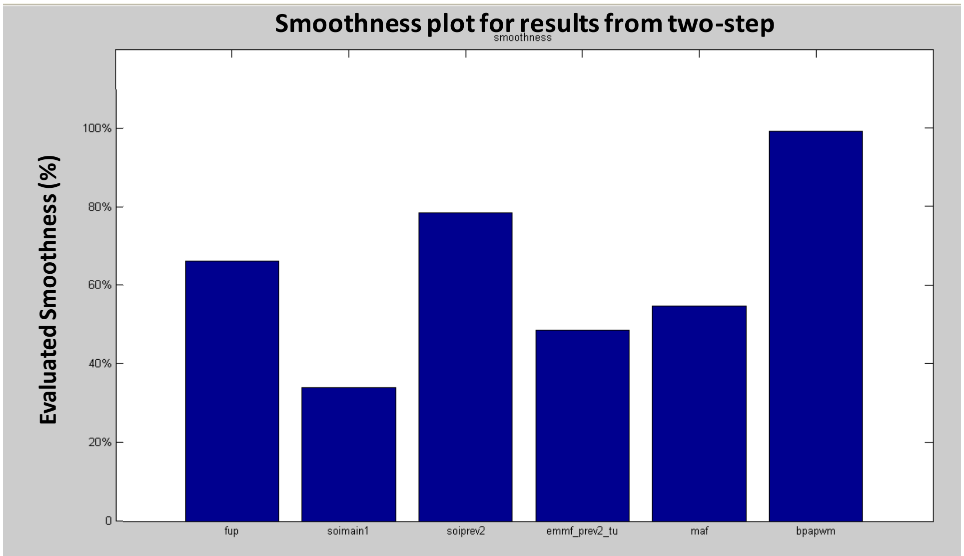
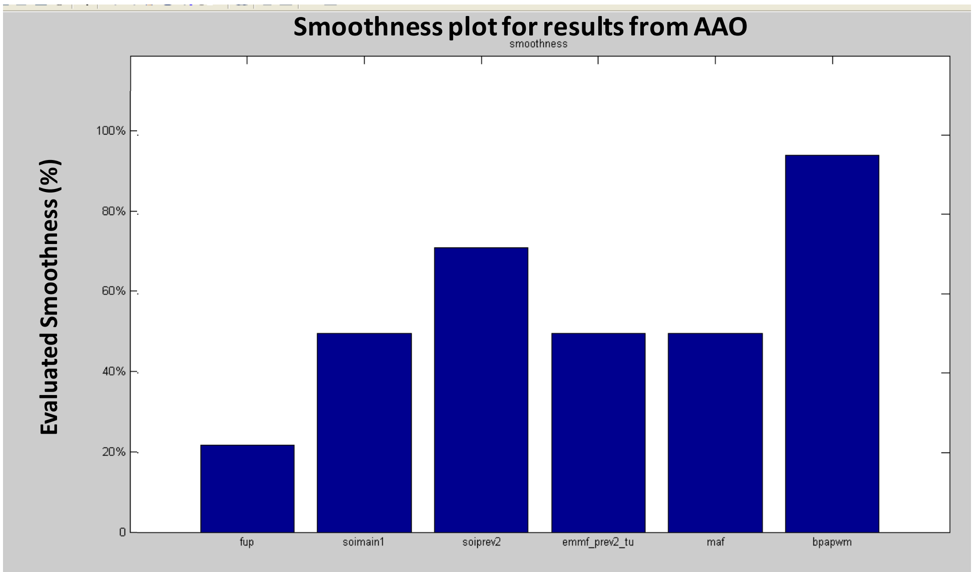
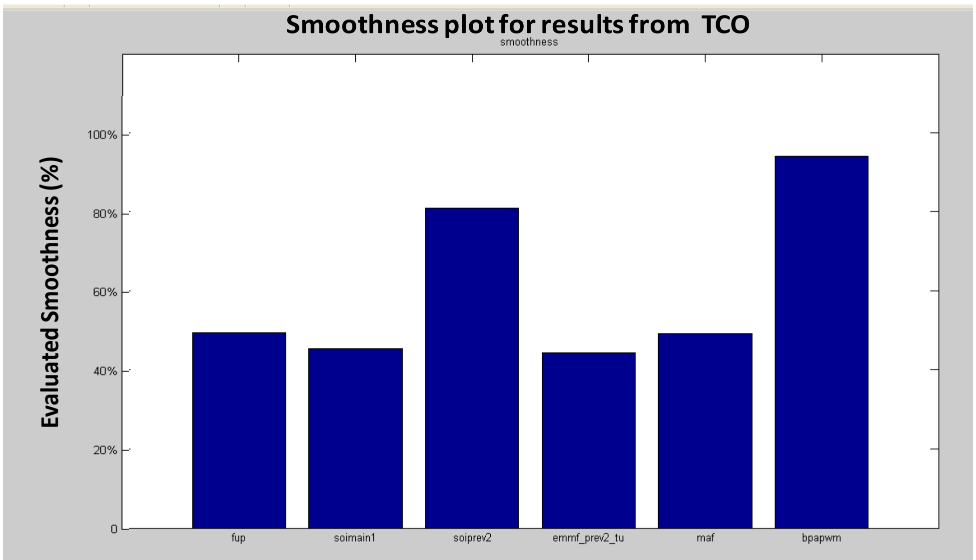


Figure 7.10 Comparison of fuel consumption for different approaches

Figure 7.11 shows a comparison of the smoothness of the optimal solution from the different approaches. The optimal solution from the two-step approach generates the worst smoothness and the optimal solution from HCO generates the best smoothness of the actuator map.


Smoothness plot for results from two-step
smoothness
Evaluated Smoothness (%)
100%
80%
60%
40%
20%
0
fup
soimain1
soiprev2
emmf_prev2_tu
maf
bpapwm


Smoothness plot for results from AAO
smoothness
Evaluated Smoothness (%)
100%
80%
60%
40%
20%
0
fup
soimain1
soiprev2
emmf_prev2_tu
maf
bpapwm


Smoothness plot for results from TCO
smoothness
Evaluated Smoothness (%)
100%
80%
60%
40%
20%
0
fup
soimain1
soiprev2
emmf_prev2_tu
maf
bpapwm

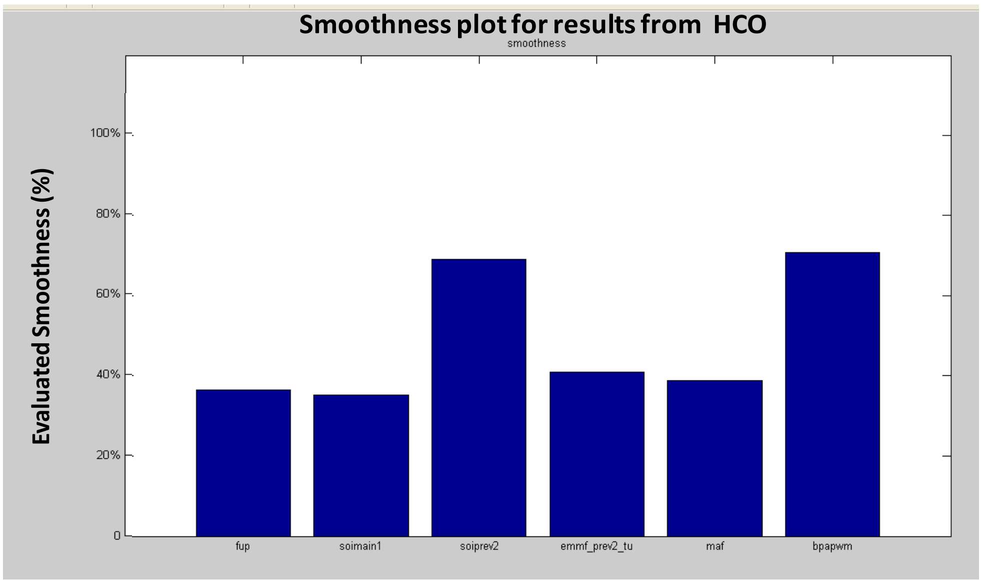


Figure 7.11 Comparison of smoothness for different approaches

Figure 7.12 illustrates the emissions performance of HCO in comparison to the solutions generated from AAO and TCO. It shows that NOx is the most difficult constraint to achieve by delivering the optimal solution of NOx on the edge of the limit.

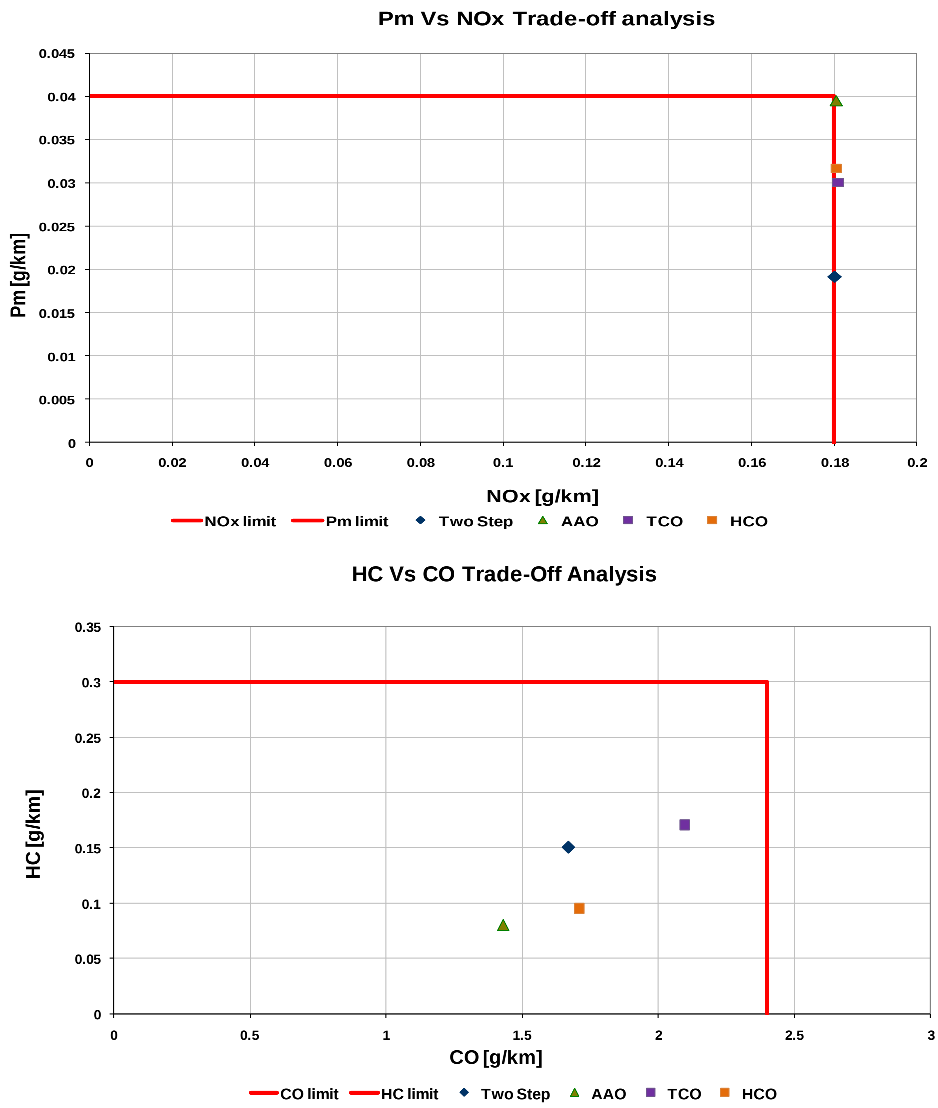
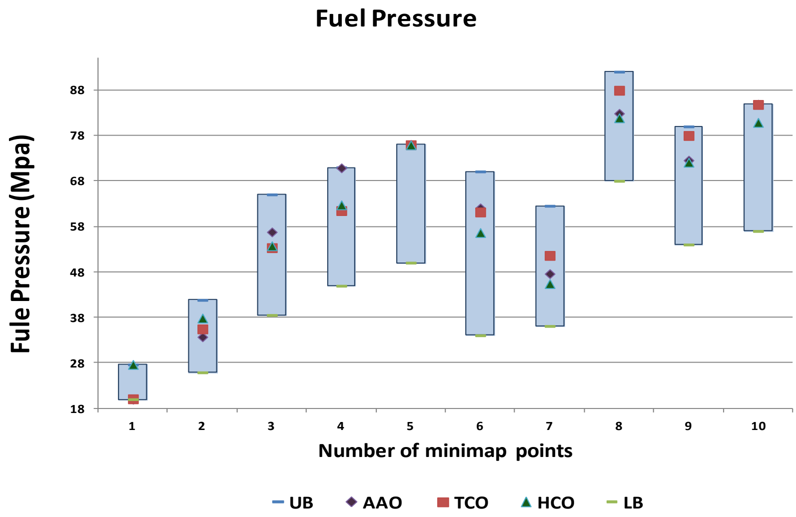
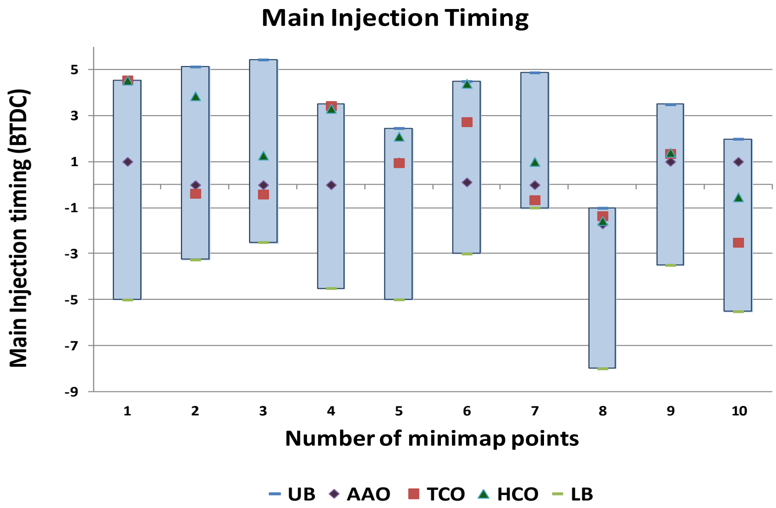
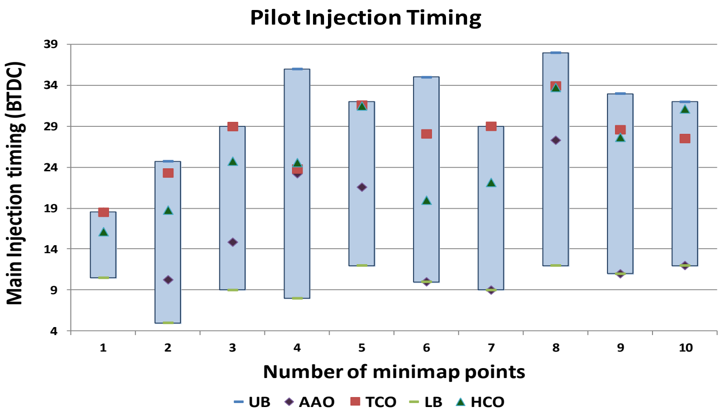


Figure 7.12 Illustration of emissions for different approaches with limits

Figure 7.13 shows the optimal solution actuator settings in the design space. The optimal solution of actuator settings from the different approaches are distributed over a wide range, which meant that the different optimisation approaches resulted in different local optimal solutions.


Fuel Pressure
Fule Pressure (Mpa)
88
78
68
58
48
38
28
18
1
2
3
4
5
6
7
8
9
10
Number of minimap points
UB
AAO
TCO
HCO
LB


Main Injection Timing
Main Injection timing (BTDC)
5
3
1
-1
-3
-5
-7
-9
1
2
3
4
5
6
7
8
9
10
Number of minimap points
UB
AAO
TCO
HCO
LB


Pilot Injection Timing
Main Injection timing (BTDC)
39
34
29
24
19
14
9
4
1
2
3
4
5
6
7
8
9
10
Number of minimap points
UB
AAO
TCO
LB
HCO

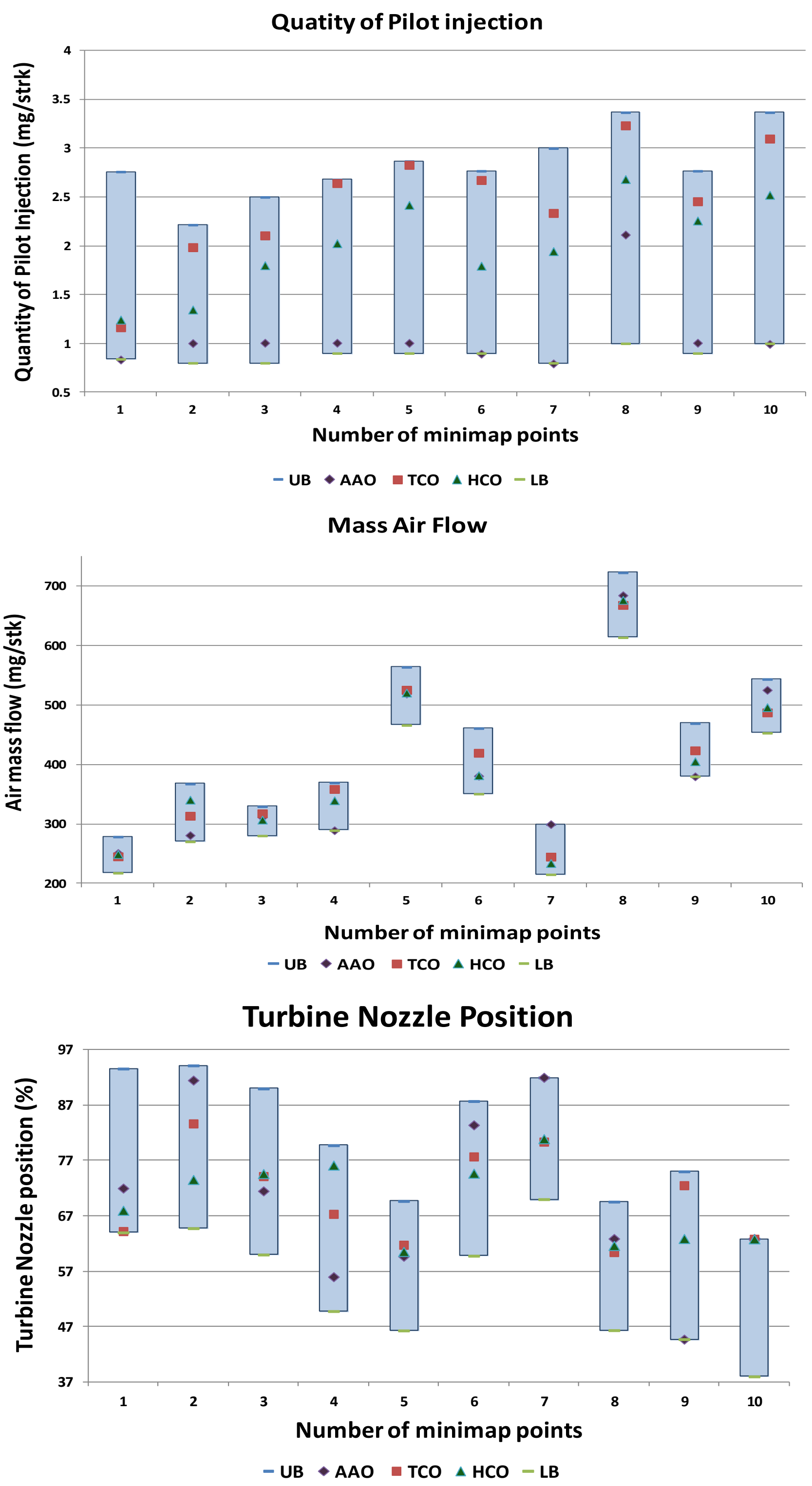


Figure 7.13 Comparison of Optimal solution actuator settings AAO, TCO and HCO with $J2_{smooth}$ formulation

7.5 Full Cycle HCO Results Performance Analysis:

In order to summarise the performance of HCO, Table 7.2 summarises and comparison with the fuel consumption, the emissions, and the computation time cost of different HCO approaches.

Table 7.2 Comparison of results between HCO formulations and Two Step approaches

Column1	NOx(Kg)	CO(Kg)	HC(Kg)	PM (Kg)	Fuel (L/100KM)	Fuel consumption improvement	Computation (hour)
Two Step	0.00198	0.018366	0.001655	0.00021	7.383		
HCO V1	0.001985	0.016763	0.001031	0.000419	7.08	↓ 4.1%	22.4
HCO V2	0.001985	0.018085	0.001067	0.000369	7.02	↓ 4.9%	40.59
EMS Limits	0.00198	0.0264	0.0033	0.00044			

Table 7.2 shows that all emissions are within legislative limits, HCO V2 and V3 significantly improved the fuel consumption, by 4.1% and 4.9%, respectively, as compared to the two-step approach.

Figure 7.14 compares the smoothness of HCO and the optimal solution from different local smoothness formulations.

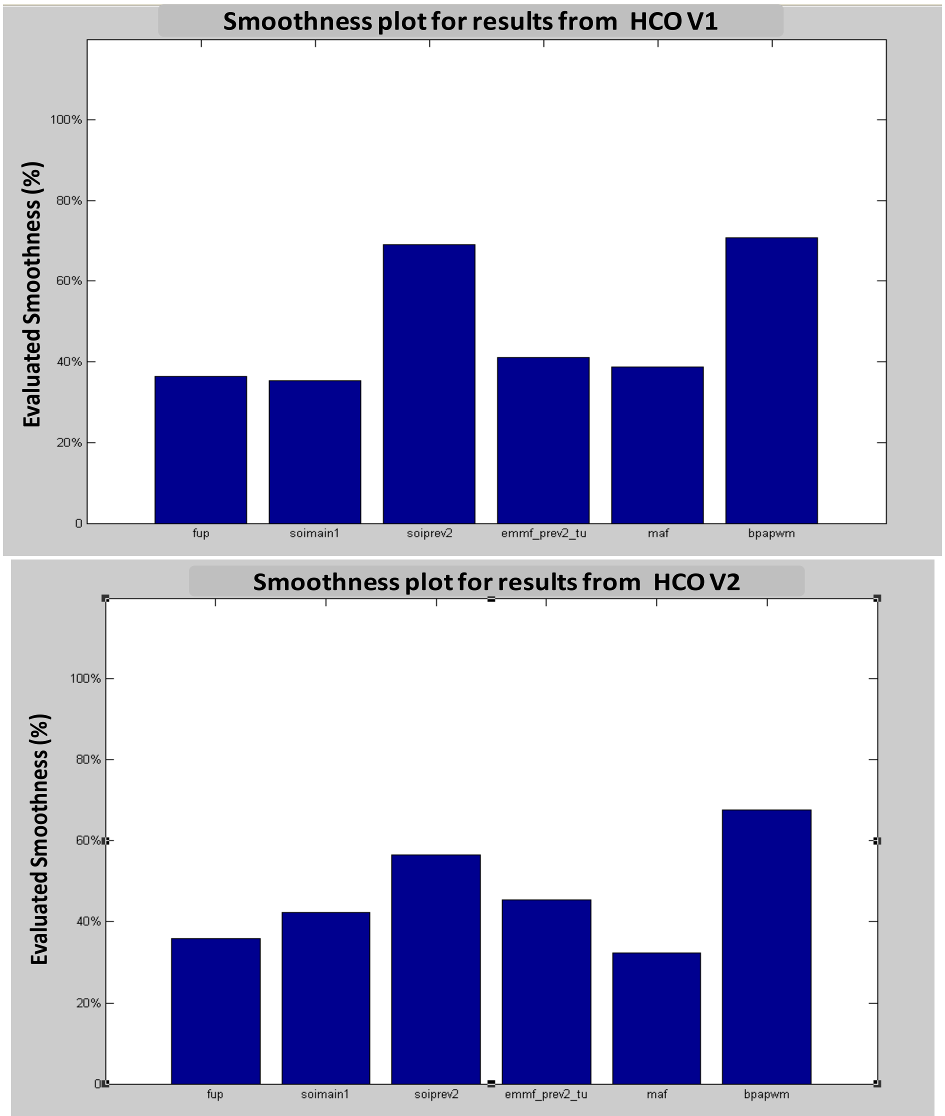


Figure 7.14 Smoothness Plots of the Optimal Solutions from Different HCO Formulations

According to the sensitivity of the engine actuator map smoothness, the fuel pressure, air mass flow and variable geometry turbine position are more sensitive than the rest actuators. For example, air mass flow is not able to change too rapidly, because the rapid increase of air mass flow required rapid boost pressure, which need time to be built up commonly known as turbo lag. Therefore, compare the smoothness of these three variables, HCO framework with the subsystem formulation of $J2_{smooth}$ has produced the better smoothness for the engine control maps. Whereas, the HCO framework with the subsystem formulation of $J1_{smooth}$ has produced the better fuel consumption. From these results, the comparison has clearly shown the trade-off between fuel economy and smoothness of an engine control map. HCO has illustrated the trade-off with different formulations.

7.6 Summary

This chapter presented a revised Collaborative Optimisation formulation of the Diesel Calibration optimisation problem, which was called Hybrid Collaborative Optimisation (HCO).

In terms of problem formulation, HCO delivers a significant enhancement over TCO by removing the actuator gradient constraints and replacing it with minimisation of the actuator change at the discipline (subsystem or minimap point) level optimisation. This makes the MDO problem formulation significantly different from the conventional CO (TCO) where the discipline optimisation problem is to minimise the discrepancy between system target and discipline feasible solution. This significantly reduces the number of constraint evaluation in TCO, leading to a significant improvement in computation cost (reduced by 50%, as shown by the results in Table 7.2).

Figure 7.11 shows that HCO delivers a significant improvement in the smoothness of the solution – by showing a considerable reduction in the maximum actuator change between minimap points.

However, in terms of fuel economy the performance of the HCO does not appear to be as good as the TCO and AAO; this is likely due to the introduction of the penalty function, and it can be expected that by tuning the penalty function a different trade-off between fuel economy and smoothness can be found.

From a practical point of view, an important advantage of HCO is that it reduces the need for prior calibrator input in terms of the actuator change constraints – which are no longer needed in the HCO formulation.

Chapter 8 Discussions

8.1 Scope of research

This project was defined in the context of a conventional steady state Diesel engine calibration process, where the engine testing data is collected at a predetermined number of engine operating points. Due to the use of engine steady state response models within the sponsoring company, this project was limited to be based on the steady state engine response models. However, the engine testing and modelling techniques have been recently developed for more advanced engine models (ref). The provided model structure was under a number of minimap points, which are representing the engine operating condition of the NEDC drive cycle. The optimisation problem has been defined based on the NEDC drive cycle to minimise the fuel consumption and satisfy the cycle emission constraints and local constraints.

The objective defined as identifying or developing a computational optimisation methodology for Diesel engine calibration problem based on the steady state engine response model. That can efficiently deliver the Diesel engine calibration optimisation task, which was minimising the cost function with high dimensional design space. Especially, the new generation Diesel engine involves more modern technologies, which lead to more control variables. Therefore, the optimisation methodology should be potentially suitable for the future Diesel engine calibration task. The studied optimisation methodology tested based on the provided testing data on the V6 Twin Turbo 2.7L Diesel engine from the sponsoring company.

This chapter presents a critical review of the methodology, developments and results in this thesis.

8.2 Critical Review of Methodology and Developments

The critical review of Diesel engine calibration process as described in the main literature sources and the process currently used by the Sponsoring Company, pointed the shortcomings are summarised below:

1. From the discussion in Chapter 5, the Diesel engine calibration optimisation problem can be described as a multi-dimensional optimisation problem, typically with more than 100 variables. In the literature (Roudenko, 2002; Lygoe, 2010; Styron, 2008), the Diesel engine calibration problem has still been handled as an optimisation problem “globally” over the drive cycle, which was minimising the weighted sum of engine emissions or fuel consumption. Whereas, the formulation of “global” optimisation over the drive cycle has to handle a large dimension of input variables.
2. On reviewing the current process used by the Sponsoring Company, it is clear that the optimisation problem formulation is not focused on the global objective of the Diesel engine calibration optimisation, i.e it does not minimise the total fuel consumption. Instead, the global solution is the ‘best’ possible combination out of a set of suboptimal solutions (typically 5) from the local PM-NOx Pareto trade-off. In the literature (Roudenko et al., 2002), It has been proved that the formulation of “weighted sum” over the drive cycle has better ability of exploring the design space, which leads to the better quality of global solution in terms of minimising the emissions or fuel consumption over the drive cycle. But, this optimisation process has to handle the difficulty of finding

feasible solution, which was cost by the heavy constraints on both design and decision space.

3. Another drawback of current process is that a solution is not guaranteed – in which case the local trade-off optimisation stage has to be repeated.
4. Similarly in the literature (Lygoe, 2010), a large number of population size (typically 20,000 to 100,000) was used to avoid not finding feasible solution when employed the Genetic Algorithm. This simply demonstrated that there is a trade-off between the computation of searching feasible solution and possibility of finding feasible solution.
5. From both the current process and the conventional process in the literature (Roudenko, 2002; Lygoe, 2010; Emtage, 2009), there are a huge amount of constraints that restrict both the searching space (design space) and . Throughout the constraints study in Chapter 5, it has shown that the engine actuator maximum allowable change constraints were the most restrictive,which were defined based on the calibration engineers' experience or from the similar engine calibration experience.

In summary, the Diesel optimisation problem has a number of issues, such as high dimensionality, heaviely constrained both design and decision space. Whereas, the optimisation formulation and searching algorithm that have been used in the past cannot solve all the issues at same time. A method is required to deal with all these issues simultanously as an entirty to improve the Diesel engine calibration process.

8.3 Discussion of Diesel Engine Calibration Optimisation Problem

As discussed in the literature review the Diesel model based steady state engine calibration process generally involves optimisation problem formulation at two level:

“global” or system level associated with the engine fuel consumption and emissions over the drive cycle, and the “local” level problem which is associated with engine operating limitations at a particular engine speed – load point, such as engine combustion stability, exhaust temperature and noise.

The optimisation literature review in Chapter 2 has shown that MDO frameworks have been introduced as a practical approach for dealing with modern engineering systems, which are increasingly complex, with high dimensionality and strong coupling interactions. The MDO frameworks have shown the benefits of that decrease the dimensionality, simplify the optimisation problem and reduce the cost of the analysis while maintaining the consistency of the whole system (Kroo et al., 1994).

While research was reported on the application of MDO/CO frameworks to automotive system design (Kim et al., 2001), there is no reported application of MDO/CO frameworks for steady state calibration optimisation problems. The engineering analysis of the Diesel engine calibration problem based on conventional steady state tests, discussed in Chapter 4, pointed to a 2 level structure – “Global” level and a “Local” level, relating to a number of local (minimap) optimisation problems. Based on this analysis, it was argued that MDO can be regarded as a natural framework for the steady state Diesel engine calibration optimisation problem, which the system optimisation problem is associated with the “global” level (i.e. performance over the drive cycle), and the “disciplines” are associated with the individual minimap point optimisation problems. This partitioning is inherent to the way steady state Diesel engine testing is conducted, and from an MDO point of view, it has the apparent advantage that the coupling between “disciplines” is not very strong in that there is no sharing of variables between minimap points. This is based on a rather simplistic assumption that the actuator settings at steady states can be treated as pseudo-independent variables, which ignores the stochastic relationships between these states (i.e. the fact that the real operation of an engine is dominated by transients). The introduction of actuator change constraints, which are in essence a formulation of calibrators' preference for smooth actuator maps (commonly associated with good drive-ability), is a way of introducing coupling variables between disciplines, which in effect attempt to deal with the shortcomings of this simplistic approximation.

It also can be argued that the Diesel engine calibration problem could be potentially partitioned in disciplines in different ways, e.g. by the global objectives, such as Fuel Consumption (FC), NOx, PM, HC and CO. In this case all the variables (actuator settings at all minimap points) would be shared between all the disciplines. Therefore, the formulation under CO framework would create a huge number of auxiliary variables in order to make the disciplines consistent, which is likely to increase the computation burden within a MDO/CO framework.

8.3.1 Discussion of MDO/AAO implementation

(Allison et al., 2005) discussed that the MDO/AAO framework is discussed as the most basic MDO framework. As shown in Figure 5.2, the Diesel engine calibration problem can easily fit the MDO/AAO framework, which handles the system/global analysis and local discipline / minimap points together.

Compared to the conventional two-stage approach to Diesel calibration optimisation, MDO/AAO has the advantage that it solves concurrently the global and the local optimisation problems, while also embedding calibrator's preference for a smooth actuator map in the optimisation problem formulation. This removes the need for detailed a posteriori analysis of the local Pareto trade-offs, and because the system optimisation is focused on the key deliverable – fuel economy, it should give a better solution.

However, the initial implementation highlighted computation difficulties associated with MDO/AAO, stemming from an over-constrained design space, which discussed in chapter 5. In order to address this difficulty a new algorithm (“Corrector” function) was developed and implemented in the GA tool to ensure the feasibility of both initial population and off-spring solutions. However, the introduction of the “corrector” GA operators has attracted severe computation penalty (more than 60 hours). From an engineering point of view this computation penalty was deemed acceptable by engine calibrators at the Sponsoring Company (equivalent to a run over the weekend), provided a competitive solution is delivered.

8.3.2 Discussion of MDO/CO implementation

In the literature review (Kroo, 2004) the MDO/CO framework is discussed as one of the most effective MDO frameworks for the complex engineering system optimisation problem with large dimensionality and coupling disciplines. As illustrated in Figure 6.1, the Diesel engine calibration problem can be naturally partitioned into a two levels Collaborative Optimisation framework. As discussed in chapter 6, at the system level the Diesel engine calibration optimisation problem was formulated as minimising the total cycle fuel consumption, while satisfying the cycle emission constraints. decomposed into a number of simpler disciplines.

The main advantage of the MDO/TCO formulation is that the actuator change constraints that were shown to considerably increase the difficulty of finding the feasible solutions are effectively removed from the global level optimisation problem, and delegated to the discipline level. Therefore, the global level (system level) optimisation is less constrained, thus reducing the problem of finding feasible solutions. In essence, the MDO/TCO framework has broken down the very complex optimisation problem down into a number of smaller optimisation tasks (local / discipline optimisation). The MDO/TCO process has shown improved computation time compared with the MDO/AAO process and more importantly ensures good convergence and quality solutions even with small population sizes (50). However, the MDO/MDO/TCO process still shows the loss of feasible population and the computing time expense still more 40 hours.

8.3.3 Discussion of MDO/HCO Implementation

To further improve the MDO/TCO framework, a hybrid CO (MDO/HCO) framework MDO/HCO has been developed and implemented. MDO/HCO completely removes the actuator change constraints and introduces a new strategy for discipline level optimisation - to optimise the local actuator "smoothness" rather than simply satisfying the actuator change constraints.

This innovative formulation of local actuator smoothness does not only reduce the complexity of the Diesel calibration optimisation problem (by removing the actuator change constraints), but provides the opportunity for calibrating the Diesel engine without the need to specify limits for actuator change between minimap points – hence reducing significantly the need for prior calibration knowledge input.

The results of the MDO/HCO implementation showed that the computation expense has been significantly reduced. Also, the issue of convergence that is cost by the restrictive consistency constraints is addressed by using the penalty function to deal with the consistency constraints. However, the overall system objective (fuel economy) of the result from MDO/HCO was not as good as the other frameworks. From another point of view, the formulation of consistency constraints analysis convinced tuning the trade off between fuel consumption and engine map smoothness.

From the implementation of MDO/TCO and MDO/HCO, it can be discussed that the lay out of Diesel engine calibration problem under CO based frameworks were very similar to the ATC framework, especially when the consistency constraints analysis was handled as a penalty function, which essentially as a part of objective at system level. Whereas, the most distinctive difference between CO and ATC is that the CO framework has the ability of handling the coupling disciplines, which the ATC has not. Also, the ATC has a clear hierarchical structure from system level to subsystem level, which there is no “state variable” (or shared variable) between system level and subsystem level, even between the disciplines at the same subsystem level. Therefore, In the case of the Diesel engine calibration studies, the strong coupled subsystems by the actuator change limits are hard to be precipitated within the ATC framework.

Compared to the MDO/AAO approach, the implementation of MDO/TCO formulation has shown the advantage that it formulates the engine calibration optimisation problem into global level and local level. Compared with the current two-stage approach commonly used in industry, the CO based framework formulations have used principally same idea, which is to partition the large optimisation problem with high dimentionality of design space. Whereas, the CO based frameworks have also maintained the global focuses, such as minimising fuel consumption and satisfying the cycle emission constraints.

8.4 Discussions of MDO implementation and results

In order to compare and discuss the MDO implementation with current process, Table 8.1 and table 8.2 summarise the optimisation results from the different process (including the current process result) based on the same engine testing data. Table 8.1 only shows the emissions, fuel consumption and computing time expense. It shows the variance of the results of fuel consumption and computing time expense of different processes, which AOO process produce the best fuel consumption and MDO/HCO spent the lest computing time. All the simulations in this research have been running on a multi-core computer with an Intel Q6600 Core 2 Quad processor 2.4GHz. Table 8.2 concludes the steepest gradient (in the ratio of neighbouring actuator change and actuator change limit) on each engine control settings, which produced by the optimal solutions from different process.

Table 8. 1 Summarised the optimisation results from the different process

Optimisation Approach	NOx(Kg)	CO(Kg)	HC(Kg)	PM (Kg)	Fuel (L/100KM)	Fuel Consumption improvement	Computation (hour)
TWO STEP	0.00198	0.018366	0.001655	0.00021	7.383		
AAO	0.001985	0.018594	0.000106	0.000438	6.217	15.8%	62.18
TCO	0.001986	0.024302	0.001524	0.000237	6.603	10.6%	41.52
HCO	0.001985	0.016763	0.001031	0.000419	7.08	4.1%	22.4
EMS Limits	0.00198	0.0264	0.0033	0.00044			

Table 8. 2 Comparison of the optimisation results for actuator change gradient

Smoothness	TWO STEP	AAO	TCO	HCO
Fuel Pressure	50%	22%	50%	35%
Main Injection Timing	34%	50%	46%	35%
Pilot Injection Timing	78%	72%	81%	64%
Pilot Injection Quantity	49%	50%	42%	44%
Mass Air Flow	50%	50%	50%	22%
Turbine Nozzle Position	94%	95%	94%	66%

Table 8.3 summarises the developed optimisation approaches for the Diesel engine calibration problem from this research. As the different key strategies have been employed to address the issues of finding feasible solution and maximise the exploration of feasibility in design space, the different approaches have demonstrated the advantages and disadvantages as summarised in Table 8.3. The comparison of the analysis different approaches, the MDO/HCO is recommend as a efficient optimisation process for the Diesel engine calibration problem. It also has shown the potential of handling a larger engine calibration problem when more minimap points are defined and more engine control variables are involved.

Table 8. 3 Summary of key strategies and developments in the MDO implementation

<table>
<tr><th>Framework</th><th colspan="2">Strategy for handling difficulty with feasibility</th></tr>
<tr><td rowspan="3">MDO/AAO</td><td colspan="2">Enhanced GA algorithm development to ensure feasible population:
- Subsidiary optimisation (“Corrector” function) to ensure feasibility of initial population;
- Subsidiary optimisation (“Corrector” function) to ensure feasibility of off-springs.</td></tr>
<tr><td>Advantages</td><td>Disadvantages</td></tr>
<tr><td>- Ensures good convergence and quality solutions even with small populations.</td><td>- Computation penalty due to subsidiary optimisation algorithms</td></tr>
<tr><td rowspan="4">MDO/TCO</td><td colspan="2">Relax constraints
- Actuator change constraints removed from system level optimisation and “delegated” to subsystem (local / discipline) optimisation;</td></tr>
<tr><td>Advantages</td><td>Disadvantages</td></tr>
<tr><td>- Improved computation time.</td><td>- Computation is still expensive due to complex subsystem optimisation.</td></tr>
<tr><td>- Ensures good convergence and quality solutions even with small populations.</td><td>- Still shows loss of feasible population, but the algorithm appears stable and robust.</td></tr>
<tr><td rowspan="5">MDO/HCO</td><td colspan="2">Remove constraints
- Actuator change constraints completely removed;
- Subsystem objective includes minimisation of actuator change.</td></tr>
<tr><td>Advantages</td><td>Disadvantages</td></tr>
<tr><td>- Improved computation time;</td><td></td></tr>
<tr><td>- Use of a penalty function to address difficulties with consistency offers the benefit of a tune-able optimisation (via changing the penalty function weights) – biased towards system or sub-system.</td><td>- Consistency harder to achieve;
- Overall system objective (fuel economy) not as good as the other frameworks.</td></tr>
<tr><td>- No need for extensive analysis a priori – gradient / actuator change constraints not required – this is a significant benefit in terms of time.</td><td></td></tr>
</table>

8.5 Limitations of this research

Due to the time limits of the research period and the scope of this research, there are a number of limitations of this research that are discussed below.

As discussed this research was concentrated on the optimisation system framework and organization, the study have been limited to the use of a specific GA toolbox. Therefore, a number of more advance Evolutionary Algorithms (i.e. Particle Swarm Optimisation algorithms) have been neglected. In the literature, the Particle Swarm Optimisation algorithms have been tested for the generic optimisation problem and proved that was more efficient compare with the Genetic Algorithms (Hassan, 2005). Since the PSO algorithm is also a population based searching algorithm, it can be neatly implemented within the MDO frameworks that have been introduced in this research.

Due to the time limitation of this project, the statistical performance and robustness of the presented MDO/TCO and MDO/HCO results have not been studied. More optimisation simulation run should be conducted in order to comparing if the optimisations can be converged to the same optimal solution area from different initial population points. And also, the sensitivity of convergence that effect by population size can be tested. Due to the time consuming of the test for each of the optimisation run and time limitation of this project, it has not been done.

In order to make a fair comparison, the third order polynomial model was fitted for the engine responses. Whereas, discussion in Chapter 4 shows that there are a number of advanced statistical modelling techniques, which can predict the engine responses more accurately. These more advanced engine models should be employed to develop the engine response model quality, which can help to improve the robustness of optimal solution by predicting the more accurate engine response values.

From the research point of view, the discontinuous convergence performance of both MDO/TCO and MDO/HCO can be studied in order to understand how the different MDO frameworks affect the optimal solution searching route.

Because of the lack of the engine data, this research was only based on the only one vehicle application of the given engine, which used in the sponsoring company. It would be useful to indentify MDO/AAO, MDO/TCO and MDO/HCO frameworks for Diesel engine calibration optimisation problem by testing and validating on the different engine or different vehicle applications.

Chapter 9 Conclusions and Recommendations for Further Work

9.1 Conclusions

The main aim of this research was to develop an efficient mathematical methodology for the Diesel engine calibration optimisation based on steady state engine test data. The work concentrated on implementing a Multidisciplinary Design Optimisation framework for the Diesel engine calibration optimisation problem.

Based on the research presented in this thesis it can be concluded that the MDO frameworks could be considered a natural choice for the engine calibration optimisation problem when data is collected at steady state points in the engine speed/load space.

The MDO frameworks provide a better organisation of the steady state engine calibration problem, as a two level ("Global" and "Local" level) structure. The MDO frameworks are flexible and expandable so they have the flexibility to accommodate requirement of the increased complexity of engine calibration.

The research also demonstrated that the MDO algorithms have the potential to deliver significantly better solutions in particular in terms of the "Global" calibration optimisation objectives (such as fuel economy) compared to the conventional two-step approach. At the same time, MDO have been shown to have the potential to significantly enhance the effectives of the process by both reducing the computation time and the requirements for upfront calibrator knowledge input.

This research has considered 2 conventional MDO frameworks / formulations, i.e. All-At-Once (MDO/AAO), Traditional Collaborative Optimisation (MDO/TCO) and has proposed a novel “Hybrid” Collaborative Optimisation (MDO/HCO) formulation.

Based on the critical review and comparison between these 3 approaches the following conclusions can be made:

- The implemented MDO/AAO framework has generated solutions with a significant improvement in main Global objective (fuel economy) while satisfying all Global constraints (cycle emissions and drive-ability / actuator change constraints) and local constraints.
- The study of the constraints revealed a very high computation difficulty of Diesel calibration optimisation problem (the “chance” of a feasible solution is less than 6×10^{-6}. This leads to severe computation difficulties with the GA based evolutionary algorithms tested.
- The strategy and algorithms developed for “forcing” feasible solutions (by using the “corrector” algorithms both for the generation of the initial populations and as an evolutionary operator) was proved to be effective in promoting an effective exploration of the design space and generating highly competitive solutions; however, this was seen to add considerable computation penalty (still competitive compared to the conventional two-step process).
- The introduction of the Collaborative Optimisation (TCO) framework has proved that the two levels MDO structure enhances the computation efficiency and generates competitive Global optimal solutions that meet all “Global” and “Local” constraints. This enhancement in effectiveness is due to a better

management of the constraints; i.e. by delegating some of the constraints to subsystem level optimisation, the system level computation difficulty is greatly reduced. This improvement in computation compared to AAO process become more obvious when the complexity of the problem was raised to full space / 10 minimap points from the reduced complexity problem of only 3 minimap poitnss considered duringt he development of the algorithm.

- The Hybrid Collaborative Optimisation (MDO/HCO) algorithm, introduced to further simplify the complexity of the Global optimisation problem by delegating the actuator change constraints to the subsystem level, was proved to be effective. This new organisation of the problem required a different approach at the discipline / subsystem optimisation – minimising actuator map gradient (via the actuator change variable) rather than the standard CO (MDO/TCO) approach of minimising the discrepancy between the discipline / subsystem and the Global / system level target.
- In order to maintain consistency of the system and ensure convergence, a penalty function approach is recommended for the “Global” / system level optimisation (combining the main optimisation objective with system consistency). This leads to a potential loss in terms of the main objective (fuel consumption) in particular when a simple additive penalty function is used (as in the implementation considered in this research). However, this delivers an important opportunity from the engineering point of view – that of a trade-off between the Global objective (fuel consumption) and the subsystem / discipline objective – a smooth actuator map. Results shown that HCO has delivered the smoothest actuator maps, but the smallest improvement in fuel economy compared with the other 2 methods. This is important because it

has the potential to deliver a steady state calibration closer to the final solution – as smooth actuator maps are better in terms of transient behaviour.

- The practical importance of HCO is that it does not require gradient change constraints to be specified a priori. This reduces quite significantly the calibration expert effort, which should now be shifted towards evaluation of the optimal solution.

While this work has only focused on Diesel calibration, all the MDO methods developed would be applicable to other similar applications – such as gasoline steady state engine calibration.

The MDO / HCO framework could also be applied to other engineering problems that can be analysed / decomposed in a similar way.

9.2 Summary of Contribution:

The following points summarise the main original contributions made by the author:

- The application of MDO to engine calibration optimisation is novel. This includes the analysis / organisation of the engine calibration problem as a Multi-Disciplinary Optimisation problem – with disciplines assigned to steady state minimap points and "system" level associated with the "over the drive cycle" performance, and application / implementation of different MDO frameworks for optimisation.
- Development and implementation in Matlab of a series of algorithms (in particular the "corrector" algorithms) to enhance the ability of the GA

evolutionary algorithm to cope with the difficulty of finding feasible solutions due to the complex constraints in both the objective and solution space.

- Formulation and implementation of a Hybrid Collaborative Optimisation MDO framework, which addresses the computational difficulties, and offers an important engineering advantage of trading off between two important high level objectives – fuel economy and drive-ability, with potential for a significant improvement of the overall calibration process.
- Demonstrate the application of the MDO frameworks suggested to a real world engineering problem, supplied by the sponsoring company.

9.3 Recommendations for Future Work

Since this research has successfully opened a new avenue of the mathematical methodology for working with the Diesel steady state engine calibration optimisation problem, there are a number recommendations for further work:

- **Further development of the MDO framework and algorithms:** Especially, the TCO process can be run with the penalty function. Also, the weight of penalty function can be varied for the HCO formulation in order to tuning the balance between fuel economy and engine actuator map smoothness. Optimisation with different population size can be done to help understand the population affect on the objective and convergence speed.
- **Algorithm development / further improvement of computation speed:** In order to make the MDO optimisation process even more efficient, the

optimisation can be developed with other Evolutionary Algorithms, such as PSO.

- **Further application:** The MDO frameworks can be use for the different type of engine calibration (Gasoline Engine) or powertrain (Hybride vehicle powertrain), which has even more complicated engineering system that can be benefitted from the MDO structure.

Reference

ALBERER, D., HIRSCH, MARKUS., DEL RE, LUIGI. (2010) A virtual references design approach for diesel engine control optimization. *Control Engineering Practice,* **In Press, Corrected Proof**.

ALLISON, J., et al. (2005) On the impact of coupling strength on complex system optimization for single-level formulations. *Proceedings of DETC'05 Long Beach, ASME Design Engineering Technical Conferences*.

ALLISON, J. T. (2004). *Complex system optimization: A review of analytical target cascading, collaborative optimization, and other formulations*. The University of Michigan.

ALLISON, J. T., et al. (2009) Optimal partitioning and coordination decisions in decomposition-based design optimization. *Journal of Mechanical Design,* **131,** 081008.

ALONSO, J. M., et al. (2007) Combining Neural Networks and Genetic Algorithms to Predict and Reduce Diesel Engine Emissions. *Evolutionary Computation, IEEE Transactions on,* **11** (1)**,** 46-55.

ANDRE, M., et al. (1995) *Driving cycles for emission measurements under European conditions.* Society of Automotive Engineers, 400 Commonwealth Dr, Warrendale, PA, 15096, USA.

ATKINSON, C., et al. (2008) Using Model-Based Rapid Transient Calibration to Reduce Fuel Consumption and Emissions in Diesel Engines.

ATKINSON, C. and MOTT, G. (2005) Dynamic model-based calibration optimization: An introduction and application to diesel engines.

ATKINSON, C. M., et al. (1998) 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.

AYEB, M., et al. (2006) Robust identification of nonlinear dynamic systems using Design of Experiment. IEEE, pp. 2321-2326.

AYEB, M., THEUERKAUF, H., WIHELM, C. & WINSEL, T (2005) D-Optimal Design of Experiments for Neural Networks: Application to Soot Emission of a Heavy Duty Diesel Engine. *3rd Conference Design of Experiments (DoE) in Engine Development.* Berlin.

BAERT, R., et al. (1999) Efficient EGR technology for future HD diesel engine emission targets. *SAE transactions,* **108** (4)**,** 381-393.

BALLING, R. J., et al. (1994) Optimization of Coupled Systems: A Critical Overview of Approaches.

BHATTI, M. A. (2000) *Practical optimization methods: with Mathematica applications.* Telos Pr.

BLUMBERG, P. (1976) Powertrain simulation: a tool for the design and evaluation of engine control strategies in vehicles. *SAE International,* **760158**.

BOEHNER, D. (1997) Common rail injection system for commercial diesel vehicles. *SAE International,* **970345** (970345).

BOGGS, P. T. and TOLLE, J. W. (1995) Sequential quadratic programming. *Acta numerica,* **4** (-1)**,** 1-51.

BRAHMI, E. H., et al. (2009) Statistical modeling and optimization for diesel engine calibration. IEEE, pp. 1770-1775.

BRAMLETTE, M. F. (1991) Initialization, mutation and selection methods in genetic algorithms for function optimization. Morgan Kaufmann, San Mateo, CA, Vol. 100, p. 107.

BRAUN, R., et al. (1996) Implementation and performance issues in collaborative optimization. *Analysis,* **10,** 1.

BRAUN, R. and KROO, I. (1995) Development and application of the collaborative optimization architecture in a multidisciplinary design environment.

BRAUN, R. and MOORE, A. (1996) Use of the collaborative optimization architecture for launch vehicle design. Citeseer.

BROOKS, T. and LUMSDEN, G. (2005) Improving Base Engine Calibrations for Diesel Vehicles Through the Use of DoE and Optimization Techniques. *SAE* San Antonio, Texas, USA.

BYRD, R., et al. (1992) An SQP augmented Lagrangian BFGS algorithm for constrained optimization. *SIAM Journal on Optimization,* **2,** 210.

CARY, M. (2003). *A Model Based Engine Calibratioon Methodology for A Port Injection Spark-Ignition Engine*. PhD. University of Bradford.

CHANDRASHEKARAPPA, P. and DUVIGNEAU, R. (2007) Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization.

CHAUDOYE, F., CASTAGNE, M., SINOQUET, D & WAHL, F (2009) Modelling Engine Operating Space for DoE Calibration Methods. *5th Design of Experiments (DoE) in Engine Development.*

COELLO, C. A. C. (1999) A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information System, An International Journal,* **3** (1)**,** 269-308.

COHON, J. L. (1985) *Multicriteria programming: Brief review and application.* New York, NY, Academic Press.

CONLEY, W. (1980) *Computer optimization techniques.* New York: Petrocelli Books.

CONLEY, W. (1984) *Computer optimization techniques.* New York: Petrocelli Books.

COSTA, J., et al. (1999) A comparison between kriging and radial basis function networks for nonlinear prediction. Citeseer, Vol. 99.

CRAMER, E., et al. (1992) On Alternative Problem Formulations for Multidisciplinary Design Optimization.

CRAMER, E. J., et al. (1994) Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization,* **4,** 754.

DAS, I. and DENNIS, J. (1998) Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization,* **8,** 631.

DAVIS, T. and LAWRANCE, A. (2000) Engine mapping: a two stage regression approach based on spark sweeps. *Statistics for Engine Optimization***,** 99-108.

DE JONG, K. and SPEARS, W. (1991) An analysis of the interacting roles of population size and crossover in genetic algorithms. *Parallel problem solving from nature***,** 38-47.

DEB, K. (2001) *Multi-objective optimization using evolutionary algorithms.* (Wiley-Interscience series in systems and optimization) Chichester: Wiley.

DEB, K. (2004) *Optimization for engineering design: algorithms and examples.* New Delhi: Rajkamal Electric Press.

DEB, K., et al. (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Springer, pp. 849-858.

DEPINCE, P., et al. (2007) Multidisciplinary and multiobjective optimization: Comparison of several methods.

DESANTES, J., et al. (2005) Multi-objective optimization of heavy duty diesel engines under stationary conditions. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* **219** (1)**,** 77-87.

DICKMAN, B. H. and GILMAN, M. J. (1989) Monte Carlo optimization. *Journal of Optimization Theory and Applications,* **60** (1)**,** 149-157.

EDWARDS, S., CLARKE, DP AND PILLEY, AD (2000) The Role of Statistics in the Engine Development Process. *Statistics for Engine Optimization, Professional Engineering Publications, Ltd, London*.

EDWARDS, S. P., et al. (1997) The Optimization of Common Rail FIE Equipped Engines Through the Use of Statistical Experimental Design, Mathematical Modelling and Genetic Algorithms. *SAE,* **970346**.

FLETCHER, R. (1987) Practical methods of optimization. 1987. *John and Sons, Chichester*.

FOGEL, D. B. and ATMAR, J. W. (1990) Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems. *Biological Cybernetics,* **63** (2)**,** 111-114.

FONSECA, C. M. and FLEMING, P. J. (1993) Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. Citeseer, Vol. 423, pp. 416–423.

FONSECA, C. M. and FLEMING, P. J. (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* **28** (1)**,** 26-37.

FORRESTER, A., et al. (2008) *Engineering design via surrogate modelling : a practical guide.* Hoboken, N.J.

Chichester: Wiley ;

John Wiley distributor.

GLIKIN, P. (1985) Fuel injection in diesel engines. *ARCHIVE: Proceedings of the Institution of Mechanical Engineers, Part D: Transport Engineering 1984-1988 (vols 198-202),* **199** (34)**,** 161-174.

GOLDBERG, D. E. (1989a) *Genetic algorithms.* Addison Wesley.

GOLDBERG, D. E. (1989b) Genetic algorithms in search, optimization, and machine learning.

GOLDBERG, D. E., et al. (1991) Genetic algorithms, noise, and the sizing of populations. *Urbana,* **51,** 61801.

GOLDSTEIN, A. and PRICE, J. (1967) An effective algorithm for minimization. *Numerische Mathematik,* **10** (3)**,** 184-189.

GOLDSTEIN, A. A. (1962) Cauchy's method of minimization. *Numerische Mathematik,* **4** (1)**,** 146-150.

GOODMAN, M. (2006a). Robust Diesel PT Emissions Calibration. Jaguar LandRover.

GOODMAN, M. (2006b) Steady State Diesel Engine Calibratiion Process. Bradford

GRIPPO, L., et al. (1986) A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis***,** 707-716.

GROVE, D. M. and DAVIS, T. (1992) *Engineering, Quality and Experimental Design.* London, Uk: Longman

GROVE, D. M., et al. (2004) Multifactor B-spline mixed models in designed experiments for the engine mapping problem. *Journal of quality technology,* **36** (4)**,** 380-391.

GUERRASSI, N. and DUPRAZ, P. (1998) A common rail injection system for high-speed, direct-injection diesel engines.

HAFNER, M. and ISERMANN, R. (2001) The use of stationary and dynamic emission models for an improved engine performance in legal test cycles.

HAINES, S., et al. (2000) The Application of an Automatic Calibration Optimization Tool to Direct-Injection Diesels. *Statistics for Engine Optimization, Professional Engineering Publishing Ltd, London*.

HAN, Q., et al. (2004) Study on Modelling Method for Common-Rail Diesel Engine Calibration and Optimization.

HEYWOOD, J. B. (1988) *Internal combustion engine fundamentals.* (McGraw-Hill series in mechanical engineering) New York; London McGraw-Hill,.

HOLLAND, J. H. (1975) *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.* Ann Arbor, Mich.: University of Michigan Press.

HOLLIDAY, T. (1995). *The Design and Analysis of Engine Mapping Experiments: A Two-Stage Approach*. PhD. University of Birmingham.

HOLLIDAY, T., et al. (1998) Engine-mapping experiments: a two-stage regression approach. *Technometrics,* **40** (2)**,** 120-126.

HORN, J. (1997) Multicriteria decision making and evolutionary computation. *Handbook of Evolutionary Computation*.

HORN, J., et al. (1994) A niched Pareto genetic algorithm for multiobjective optimization. In: *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on.* pp. 82-87 vol.1.

HOUCK, C. R., et al. (1995) A genetic algorithm for function optimization: A Matlab implementation. *NCSU-IE TR,* **95** (09).

HOWLETT, R., et al. (1999) Neural network techniques for monitoring and control of internal combustion engines.

HUHTALA, K. and VILENIUS, M. (2001) Study of a Common-Rail Fuel Injection System. *SAE Technical Paper***,** 01-3184.

JANKOVIC, M. and MAGNER, S. (2004) Optimization and scheduling for automotive powertrains. In: *American Control Conference, 2004. Proceedings of the 2004.* Vol. 5, pp. 4054-4059 vol.5.

JANKOVIC, M. and MAGNER, S. (2006) Fuel economy optimization in automotive engines. In: *American Control Conference, 2006.* p. 6 pp.

JEONG, S., et al. (2006) Optimization of combustion chamber for diesel engine using kriging model. *Journal of Fluid Science and Technology,* **1** (2)**,** 138-146.

KAGESON, P. (1998) Cycle-beating and the EU Test Cycle for Cars. *European Federation for Transport and Environment. T&E,* **98** (3).

KAJI, H. and KITA, H. (2007) Acceleration of Experiment-Based Evolutionary Multi-objective Optimization of internal-combustion engine controllers using fitness estimation. IEEE, pp. 1777-1784.

KIM, H. M. (2001). *Target cascading in optimal system design*. The University of Michigan.

KIM, H. M., et al. (2001) Analytical target cascading in automotive vehicle design. pp. 661-670.
KODIYALAM, S. and CENTER, L. R. (1998) *Evaluation of methods for multidisciplinary design optimization (MDO), Phase I.* Citeseer.
KROO, I. (2004) Distributed multidisciplinary design and collaborative optimization. *VKI lecture series on Optimization methods & Tools for Multicriteria/Multidisciplinary Design*, 22.
KROO, I., et al. (1994) Multidisciplinary optimization methods for aircraft preliminary design. *AIAA paper*, 94-4325.
KROO, I. and MANNING, V. (2000) Collaborative optimization: status and directions. Vol. 6, p. 8.
LANGOUET, H., et al. (2008) Optimization for engine calibration.
LEE, B. (2005). *Methodology for rapid static and dynamic model-based engine calibration and optimization*. Doctor of Philosophy. Ohio State University.
LYGOE, R. J. (2010). *Complexity Reduction in High-Dimensional Multi-Objective Optimisation*. the University of Sheffield.
MICHELENA, N., et al. (1999) A system partitioning and optimization approach to target cascading. Citeseer.
MILLER, B. L. and SHAW, M. J. (1996) Genetic algorithms with dynamic niche sharing for multimodal function optimization. IEEE, pp. 786-791.
MINAMI, T., et al. (1995) Reduction of diesel engine NOx using pilot injection. *SAE Special Publications*, 89-96.
MONAGHAN, M. (1998) Particulates and the diesel-the scale of the problem. *Diesel engines-particulate control: 23 November 1998, IMechE Headquarters, London, UK*.
MONTGOMERY, D. C., et al. (2001) *Engineering statistics.* New York: John Wiley.
MOON, J. F. (1974) *Rudolf Diesel and the diesel engine.* (Pioneers of science and discovery) Hove: Wayland Publishers.
MORTON, T. M. (2002) Radial basis functions for engine modelling. Wiley, p. 43.
NARAYANAN, A., et al. (2007) Simultaneous model building and validation with uniform designs of experiments. *Engineering Optimization,* **39** (5), 497-512.
NE LER, A., SCHAUM,S., KNAAK, M., R PKE, K. (2007) Fast basic combustion engine calibration with "Rapid Measurement". In: *4th Conference Design of Experiments (DoE) in Engine Development.* Berlin: Vol. 29.
OMRAN, R., et al. (2007) Genetic Algorithm for Dynamic Calibration of Engine's Actuators.
ONDER, C. H. and GEERING, H. P. (1993) Model-based multivariable speed and air-to-fuel ratio control of an SI engine. *SAE transactions,* **102,** 1142-1142.
ONDER, C. H. and GEERING, H. P. (1995) *Model-based engine calibration for best fuel efficiency.* Society of Automotive Engineers, 400 Commonwealth Dr, Warrendale, PA, 15096, USA.
PAPALAMBROS, P. Y. (2001) Analytical target cascading in product development. Harrogate, pp. 9-10.
PAPALAMBROS, P. Y. (2002) The optimization paradigm in engineering design: promises and challenges. *Computer-Aided Design,* **34** (12), 939-951.
PAPALAMBROS, P. Y. and WILDE, D. J. (2000) *Principles of optimal design : modeling and computation.* Cambridge: Cambridge University Press.
PARK, C., et al. (2004) Effects of multiple injections in a HSDI diesel engine equipped with common rail injection system. *SAE Technical Paper*, 01-0127.

PETER, J., et al. (2007) Comparison of surrogate models for the actual global optimization of a 2D turbomachinery flow. World Scientific and Engineering Academy and Society (WSEAS), pp. 46-51.
PIENAAR, S. W. (2007). *Simulation based calibration of turbo-charger boost control*. Stellenbosch: University of Stellenbosch.
PILLEY, A., et al. (1994) Design of experiments for optimization of engines to meet future emissions targets.
RAO, S. S. (2009) *Engineering Optimization: Theory and Practice.* (Wiley-Interscience series in systems and optimization) New Jersey, US: Wiley.
RASK, E. and SELLNAU, M. (2004) Simulation-based engine calibration: Tools, techniques, and applications. *SAE SP*, 21-32.
RAVINDRAN, A., et al. (2006) *Engineering optimization : methods and applications.* Chichester: Wiley.
REKLAITIS, G., et al. (1983a) *Engineering optimization : methods and applications.* N Y: Wiley.
REKLAITIS, G. V., et al. (1983b) *Engineering optimization : methods and applications.* New York; Chichester: Wiley.
RISHAVY, E. A., HAMILTON, S. C., AYERS, J. A. AND KEANE, M. A (1977) Engine Control Optimization for Best Fuel Economy with Emission Constraints. *SAE* **770075**.
ROPKE, K. (2005) *DoE - Design of Experiments Methods and applications in engine development.* by SV Corporate Media GmbH.
ROUDENKO, O., et al. (2002) Application of a Pareto-based evolutionary algorithm to fuel injection optimization. In: *International Conferene On Statistics And Analytical Methods in Automotive Engineering* Professional Engineering Publishing, Vol. 4, pp. 81-92.
RUDOLPH, G. (1996) Convergence of evolutionary algorithms in general search spaces. IEEE, pp. 50-54.
RUDOLPH, G. (1999) *Evolutionary search under partially ordered fitness sets.* Citeseer.
SAMPSON, D. (2010) Reducing Emissions Using Model-Based Calibration in the Test Cell. In: *MathWorks Automotive Conference '10.* Stuttgart, Germany.
SAMPSON, D., AND SHERIDAN, LAD, (2004) Engine Calibration: A Model-Based Approach.
SAMPSON, D. A. S., L, (2007) Recent Developments in Techniques for Model-Based

Calibration. In: *4th Design of Experiments (DoE) in Engine Development.* Berlin, pp. 45-55.
SAMUEL, S., et al. (2002) Automotive test drive cycles for emission measurement and real-world emission levels-a review. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* **216** (7)**,** 555-564.
SAUNDERS, R. L. O. (2004). *Investigation into a Validation Methodology for Non-Parametric Models for engine Mapping*. MSc. University of Bradford.
SCHAFFER, J. D. (1985a) Multiple objective optimization with vector evaluated genetic algorithms. L. Erlbaum Associates Inc., pp. 93-100.
SCHAFFER, J. D. (1985b). *Some experiments in machine learning using vector evaluated genetic algorithms*.

SEABROOK, J. (2000) *Practical Implementation of Design of Experiments in Engine Development. Statistics for Engine Optimization.* Edwards SP, Grove DM & Wynn HP Professional Engineering Publishing.
SEABROOK, J., CORNWELL, R., MORTIMER, P. AND EDWARDS, S. (2007) A Calibration Methodology to Ensure Low Variation in Emissions between Engine in Production. *4th Conference Design of Experiments (DoE) in Engine Development.* Berlin.
SHERIDAN, L. A. D., GOYDER, R., CHERRIE, J. B. AND MORTON, T. M. (2004) Defining a model-based calibration process for a twin-independent valve timing engine. In: *Control Applications, 2004. Proceedings of the 2004 IEEE International Conference on.* Vol. 2, pp. 1431-1436 Vol.2.
SINOQUET, D., LANGOUET, H., CHAUDOYE, F. AND CASTAGNE, M. (2009) Multi-Objective Constrained Optimisation of Engine Maps. *5rd Design of Experiments (DoE) in Engine Development.* Berlin.
SMITH, A., et al. (1998) Effect of engine tune on the emissions performance of a modern high speed direct injection diesel engine. *SAE SPEC PUBL, SAE, WARRENDALE, PA,(USA), Feb 1998,* **1326,** 21-32.
SOBIESZCZANSKI-SOBIESKI, J., et al. (1998) *Bi-level integrated system synthesis (BLISS).* Citeseer.
SOBIESZCZANSKI-SOBIESKI, J. and HAFTKA, R. T. (1997) Multidisciplinary aerospace design optimization: survey of recent developments. *Structural and Multidisciplinary Optimization,* **14** (1)**,** 1-23.
SOBIESZCZANSKI-SOBIESKI, J. and KROO, I. (1996) Collaborative optimization applied to an aircraft design problem. *AIAA paper***,** 96-0715.
SRINIVAS, M. and PATNAIK, L. M. (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on,* **24** (4)**,** 656-667.
SRINIVAS, N. and DEB, K. (1994) Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation,* **2** (3)**,** 221-248.
STONE, R. (1999) *Introduction to internal combustion engines.* London: Macmillan.
STONE, R. and BALL, J. K. (2004) *Automotive engineering fundamentals.* Warrendale, Pa.: SAE International.
STOTZ, M., et al. (2000) Potential of Common-Rail Injection System for Passenger Car Di Diesel Engines. *SAE International,* **2000-01-0944**.
STUHLER, H., et al. (2002) Automated model-based GDI engine calibration adaptive online DoE approach. *SAE Technical Paper***,** 01-0708.
STUMPP, G. and RICCO, M. (1996) Common rail, an attractive fuel injection system for passenger car DI diesel engines. *SAE International,* **960870** (960870).
STYRON, J. P. (2008) Using Model-Based Calibration Toolbox Multimodels for Cycle-Optimized Diesel Calibration. In: *MathWorks Automotive Conference.* Stuttgart, Germany.
TAPPETA, R. and RENAUD, J. (1997) Multiobjective collaborative optimization. *Journal of Mechanical Design,* **119,** 403.
TETSUI, T. and ONO, S. (1999) Endurance and composition and microstructure effects on endurance of TiAl used in turbochargers. *Intermetallics,* **7** (6)**,** 689-697.
TZIRAKIS, E., et al. (2006) Vehicle emissions and driving cycles: Comparison of the Athens driving cycle (ADC) with ECE-15 and European Driving Cycle (EDC). *Global NEST Journal,* **8** (3)**,** 282-290.

VOSSOUGHI, G. and REZAZADEH, S. (2005) Optimization of the calibration for an internal combustion engine management system using multi-objective genetic algorithms. IEEE, Vol. 2, pp. 1254-1261.

VOSSOUGHI, G. R. and REZAZADEH, S. (2004) Development of an Integrated Computer Environment for Optimization of Engine Management System Calibration.

WARD, M., et al. (2002) Bayesian statistics in engine mapping. pp. 3-16.

WARD, M. C., BRACE, C. J., VAUGHAN, N. V. & KENNEDY, G. (2004) Automated limit space characterization and mapping. *SAE 2004 World Congress and Exhibition.* Detroit, Michigan, USA, SAE International, Warrendale, Pennsylvania, USA.

WATERS, W. (1972) General purpose automotive vehicle performance and economy simulator. *SAE International,* **720043**.

WEILE, D. S. and MICHIELSSEN, E. (1997) Genetic algorithm optimization applied to electromagnetics: A review. *Antennas and Propagation, IEEE Transactions on,* **45** (3)**,** 343-353.

WU, B., et al. (2006) Cam-phasing Optimization Using Artificial Neural Networks as Surrogate Models- Fuel Consumption and NOx Emissions.

YOKOMURA, H., et al. (2004) Transient Egr Control for a Turbocharged Heavy-Duty Diesel Engine. *SAE Technical Paper***,** 01-0120.

ZITZLER, E., et al. (2001) *SPEA2: Improving the strength Pareto evolutionary algorithm.* Citeseer.

ZITZLER, E. and THIELE, L. (1998) *An evolutionary algorithm for multiobjective optimization: The strength Pareto approach.* Citeseer.

ZWEIRI, Y. H. and SENEVIRATNE, L. D. (2007) Diesel Engine Indicated Torque Estimation Based on Artificial Neural Networks. In: *Computer Systems and Applications, 2007. AICCSA '07. IEEE/ACS International Conference on.* pp. 791-798.

APPENDIX I Engine Testing Data Example (750 rpm – 30Nm)

test	Date	Time	A_SP_DY	A_TRQ_Ol	embpa_pw	emfup	emmaf	emmf_tot	emsoi_main	emsoi_prev2_dif	emti_prev2_tu	Apollo_OA	Smoke1FS	Opacity	ECO_VCR	ENO_VHR	ETHC_VHI	TK_O_SU	Cyl1IMEP	Calculated
point			rpm	Nm	%	MPA	mg/stk	mg/stk	degATDC	degATDC	ms	dBA	FSN	%	ppm	ppm	ppm	deg C		mg/stk
2	3/24/2005	10:45:43	750.067	29.65	64.685	20.074	229.264	6.637	-4.922	10.5	0.257	56.931	0.8	2.88	916.38	42.719	103.102	80.517	1.754	
3	3/23/2005	9:03:36	749.991	30.353	81.219	25.513	252.608	6.788	2.391	15.188	0.221	54.949	0.45	1.81	964.994	44.54	158.755	72.562	1.909	6.977862
4	3/23/2005	9:08:25	749.504	30.325	68.823	23.475	229.887	7.032	-0.023	12	0.243	54.953	0.9	3.5	1175.311	33.618	195.758	68.48	2.001	7.115817
5	3/24/2005	10:50:20	750.447	29.92	82.721	21.46	262.989	6.648	-2.366	15.982	0.262	56.817	0.63	2.35	976.569	47.807	113.666	73.329	1.917	
6	3/23/2005	9:18:55	750.577	29.644	72.004	26.578	245.17	6.967	-4.594	13.004	0.234	57.201	0.42	1.67	1180.759	56.017	140.857	61.937	1.914	6.883593
7	3/23/2005	9:23:42	749.683	30.03	85.107	24.599	277.62	7.041	2.263	15.872	0.251	54.991	0.42	1.42	1032.795	50.687	193.415	60.219	2.082	7.158581
8	3/23/2005	9:28:30	749.902	29.241	74.148	22.717	257.589	6.638	0.023	13.617	0.274	55.299	0.69	2.78	1105.071	47.501	213.817	58.184	1.927	6.934239
9	3/24/2005	10:55:09	749.957	29.28	85.907	20.803	229.661	6.789	-2.203	17.273	0.287	55.683	1.03	3.87	1540.72	24.45	212.252	69.84	1.818	
11	3/23/2005	9:48:20	750.022	29.588	71.141	26.355	257.253	6.944	-4.662	14.25	0.239	58.239	0.41	1.58	719.217	103.389	84.918	79.759	1.919	6.799801
13	3/23/2005	9:57:45	749.378	30.449	74.661	22.121	271.702	7.035	0.249	14.921	0.287	56.173	0.71	2.84	828.04	61.424	138.562	69.275	1.992	7.028051
14	3/23/2005	10:02:33	750.264	29.835	88.842	27.344	251.888	7.025	-2.077	12.094	0.252	57.347	0.43	1.37	815.475	61.862	115.93	66.415	2.037	6.975322
15	3/23/2005	10:07:21	750.39	29.778	77.057	25.299	230.973	6.81	-4.336	15.352	0.267	57.358	0.42	1.86	1490.396	45.815	168.919	62.39	2.031	6.974151
16	3/23/2005	10:13:14	750.723	29.201	90.204	23.364	263.028	7.22	2.553	12.961	0.291	55.825	0.63	2.33	920.443	51.117	187.303	59.847	2.003	6.971058
161	3/24/2005	7:01:26	750.365	30.433	74.271	27.128	263.389	6.867	2.213	11.644	0.342	62.068	0.35	1.14	441.737	108.873	66.93	80.557	1.81	6.841115
162	3/24/2005	7:06:16	749.643	30.012	90.048	24.908	243.167	6.882	-0.234	16.078	0.363	58.721	0.73	2.6	820.086	57.626	94.945	72.786	1.956	6.981101
163	3/24/2005	7:11:05	750.537	29.66	77.444	22.811	224.634	7.012	-2.58	12.91	0.397	57.173	1.62	6.75	963.529	44.727	119.2	68.905	1.854	6.839547
164	3/24/2005	13:44:35	750.102	29.598	91.229	20.994	262.344	6.178	4.008	15.891	0.427	57.582	0.82	3.35	912.457	32.505	171.989	58.558	2.143	
165	3/24/2005	7:20:43	750.447	30.173	79.553	25.997	235.764	6.836	2.109	13.688	0.361	58.475	0.89	3.61	941.814	43.522	136.906	62.487	1.82	6.929204
166	3/24/2005	7:25:20	750.018	29.937	92.661	24.046	266.068	6.947	-0.164	16.664	0.386	57.604	0.62	2.43	960.689	68.263	122.484	60.879	1.965	7.022054
167	3/24/2005	7:30:10	749.601	30.072	81.326	22.122	248.017	6.855	-2.438	14.25	0.418	57.031	1.12	4.65	1096.183	53.73	151.995	59.985	1.929	7.02596

APPENDIX II Actuator Gradient Change Constraints

Start point	stop point	fup dec/inc		soi_main1 dec/inc		soi_prev2 dec/inc		mf_prev2 dec/inc		maf dec/inc		bpa dec/inc	
1	2	-2	15.75	-5.37	5.37	-8.25	8.25	-0.2	0.85	-160	160	-19.5	19.5
1	3	-2	36.75	-5.39	5.39	-12.25	12.25	-0.2	0.85	-120	120	-26	26
1	7	-2	31.5	-5.31	5.31	-11.5	11.5	-0.2	1.15	-48.5	48.5	-20	20
2	3	-2	26.5	-4.38	4.38	-14	14	-0.2	0.9	-61	61	-24	24
2	6	-2	29.25	-4.57	4.57	-17	17	-0.2	1.15	-130.5	130.5	-21.6	21.6
2	7	-2	23.25	-4.19	4.19	-13.25	13.25	-0.2	1.2	-155	155	-18	18
3	4	-2	21.2	-5.47	5.47	-13.45	13.45	-0.2	0.85	-49.5	49.5	-27.5	27.5
3	5	-2	33.45	-7.27	7.27	-12.55	12.55	-0.2	1.2	-285	285	-31.5	31.5
3	6	-2	20.95	-4.35	4.35	-13.45	13.45	-0.2	1.1	-132	132	-29.7	29.7
3	7	-16.5	2	-3.9	3.9	-10.75	10.75	-1.15	0.2	-115	115	-20.5	20.5
3	9	-2	28.3	-5.26	5.26	-12.08	12.08	-0.2	1	-134	134	-33	33
4	7	-24.45	2	-5.39	5.39	-14.2	14.2	-1.15	0.2	-145	145	-36	36
4	9	-2	19.6	-4.08	4.08	-12.73	12.73	-0.2	1	-119.5	119.5	-20.5	20.5
4	10	-2	26.9	-5.9	5.9	-13.45	13.45	-0.2	1.2	-241.5	241.5	-28	28
5	6	-32.7	2	-6.43	6.43	-13.55	13.55	-1.2	0.2	-153	153	-34.2	34.2
5	8	-2	28.3	-6.18	6.18	-15.2	15.2	-0.2	1.35	-249	249	-17.5	17.5
5	9	-19.6	2	-5.3	5.3	-10.57	10.57	-1.2	0.2	-166.5	166.5	-16.5	16.5
5	10	-2	19.7	-4.03	4.03	-10.15	10.15	-0.2	1.1	-62	62	-21.5	21.5
6	9	-2	29.8	-4.42	4.42	-13.18	13.18	-0.2	1.1	-67.5	67.5	-38.7	38.7
8	10	-22.6	2	-6.2	6.2	-15.35	15.35	-1.45	0.2	-265	265	-26.5	26.5
9	10	-2	20.3	-5.4	5.4	-11.42	11.42	-0.2	1.2	-145.5	145.5	-22.5	22.5

APPENDIX III All At Once Main Matlab Script

```
clear

clc

% load in the data based inputs
name = uigetfile;
load (name);
%user input information
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';
prompt={'number of nodes (engine testing point)',...
        'number of population'};
def = {'0','0'};
inputdat = inputdlg(prompt,'user difinition',1,def,options);

numb_node = str2double(inputdat{1,1});
PopSize = str2double(inputdat{2,1});
%numb_ac = str2double(inputdat{3,1});
%nvars = numb_ac*numb_node;
nvars=0;

if numb_node < 10
    options.Resize='on';
    options.WindowStyle='normal';
    options.Interpreter='tex';

prompt={'nodes1:','nodes2:','nodes3:','nodes4:','nodes5:','nodes6:','nodes7
:','nodes8:','nodes9:','nodes10:'};
    def = {'','','','','','','','','',''};
    NN = inputdlg(prompt,'selecting node number',1,def,options);
    NN = cell2mat(NN);
    NN = str2num(NN);
    % sort the actuator boundary into one array by combining the actuator
from
    % different nodes
    lb = [];
    ub = [];
    NumofInput=[];

    for i=1:numb_node
        bounds=Inputs_Lim{NN(i,1)};
        lb=[lb bounds(1,:)];
        ub=[ub bounds(2,:)];
        nvars=nvars+size(bounds,2);
        NumofInput=[NumofInput size(bounds,2)];
    end
else
    NN=[1; 2; 3; 4; 5; 6; 7; 8; 9; 10];
    lb=[];ub=[];NumofInput=[];
    for i=1:numb_node
        bounds=Inputs_Lim{NN(i,1)};
        lb=[lb bounds(1,:)];
        ub=[ub bounds(2,:)];
        nvars=nvars+size(bounds,2);
        NumofInput=[NumofInput size(bounds,2)];
    end
```

```
end
% sort the gradient constraints in the orgnised cell
GCS=cell(size(nodes,2),size(nodes,2));
for i=1:size(GC,1)
    row=GC(i,1);
    column=GC(i,2);
    GCS{row,column}=GC(i,3:end);
%     GCS{column,row}=GC(i,3:end);
end
GC=GCS;

msgtxt = {'create the initial population','or supply a set of initial
population'};
msgtit = 'uploading exam result......';
in_population = questdlg(msgtxt,msgtit,'Yes','No','load in','Yes');

switch in_population
    case 'Yes'
        initialPop = creator(lb,ub,PopSize,nvars);
    case 'No'
        initialPop=[];
    case 'load in'
        name = uigetfile;
        InPop=load (name);
        initialPop=InPop.Final_pop;
end
PopSize1=50;
PopSize2=100;
PopSize3=200;
PopSize4=500;

Nonlcon=@(x)con_sys(x,models,mm_n,res_time,Inputs_Lim,LC,Lim,GC,NN,...
    numb_node,NumofInput);

options=gaoptimset( 'PopulationSize',PopSize2,...
                    'SelectionFcn',{@selectiontournament,2},...
                    'Generations',10,...
                    'InitialPopulation',initialPop,...
                    'StallGenLimit',4,...
                    'MutationFcn',{@mutationadaptfeasible},...
                    'CrossoverFcn',{@crossoverheuristic,0.8},...
                    'creationFcn',@gacreationuniform,...
                    'Crossoverfraction',0.60,...
                    'Elitecount',2,...
                    'Tolfun',0.005,...
                    'Tolcon',0.005,...
                    'PlotFcns',{@gaplotbestf,@gaplotstopping,...
                                @gaplotscores});
[x,fval,reason,output,population,scores]=ga({@objectsys,models,res_time,...
    mm_n,Inputs_Lim,NN,NumofInput,LC,GC,numb_node},...
    nvars,[],[],[],[],lb,ub,Nonlcon,options);

final_results_check;
```

APPENDIX IV Constraints Function Matlab Script

Contents

- total 'global' cycle emission constraints check
- Gradient constraints
- local emission constraints
- combine all the different type of constraints


```
function
[c,ceq]=con_sys(x,models,mm_n,res_time,Inputs_Lim,LC,Lim,GC,NN,numb_node,Nu
mofInput)

c=[];ceq=[];

SX=cell(1,10);

loccon=[]; cyclcon=[]; acc_con=[];

Xcounter=[];

for i=1:size(NN,1)

    SX{NN(i,1)}=x(:,(size(Xcounter,2)+1):(size(Xcounter,2)+NumofInput(i)));

    Xcounter=[Xcounter
x(:,(size(Xcounter,2)+1):(size(Xcounter,2)+NumofInput(i)))];

end

SX{1,1}=[SX{1,1}(1,:) 69];
```

```
Input argument "NN" is undefined.



Error in ==> con_sys at 7

for i=1:size(NN,1)
```

total 'global' cycle emission constraints check

```
cyclcon = total_emission_check (SX,NN,numb_node,models,mm_n,res_time,Lim);
```

Gradient constraints

```
acc_con = gradient_check (SX,GC,NN);
```

local emission constraints

```
for i=1:numb_node

    mode = models{NN(i,1)};

    Lim_local = LC{NN(i,1)};

    loccon_em = Local_Emission_check(SX{NN(i,1)},mode,Lim_local);

    loccon = [loccon loccon_em];

end
```

combine all the different type of constraints

```
c=[loccon cyclcon acc_con];
```

APPENDIX V Function corrector Matlab script

```
function Final_pop =
corrector(initialPop,models,mm_n,res_time,lb,ub,LC,Lim,GC,NN,numb_node,Numo
fInput)

% matlabpool

Final_pop=[];

for i=1:size(initialPop,1)

    X0=initialPop(i,:);

    objt = @(x)correct_movement(x,X0);


Nonlcon=@(x)correct_con(x,X0,models,mm_n,res_time,LC,Lim,GC,NN,numb_node,Nu
mofInput);

    options=optimset('Display','final',...

                    'TolFun',0.0001,...

                    'Algorithm','active-set',...

                    'TolCon',0.0001,...

                    'TolX',0.0001);

    [Xsub,fval]=fmincon(objt,X0,[],[],[],[],lb,ub,Nonlcon,options);

    Final_pop(i,:)=Xsub;

end

% matlabpool close
```

```
Input argument "initialPop" is undefined.
```

Published with MATLAB® 7.6

APPENDIX VI AAO and TCO System objective Function Matlab Script

```
function
[obj]=objectsys(x,models,res_time,mm_n,Inputs_Lim,NN,NumofInput,LC,GC,numb_
node)

SX=cell(1,10);

Xcounter=[];

for i=1:size(NN,1)

    SX{NN(i,1)}=x(:,(size(Xcounter,2)+1):(size(Xcounter,2)+NumofInput(i)));

    Xcounter=[Xcounter
x(:,(size(Xcounter,2)+1):(size(Xcounter,2)+NumofInput(i)))];

end

SX{1,1}=[SX{1,1}(1,:) 69];

obj=0;

for i=1:size(NN,1)

    mode=models{NN(i,1)};

    f=mm_n(NN(i,1)).*res_time(NN(i,1)).*mode.emmf(SX{NN(i,1)})./20;

    obj=obj+f;

end



obj=obj*100/((1e+6)*0.84*11); %unit in L/100Km
```

```
Input argument "NN" is undefined.


Error in ==> objectsys at 4

for i=1:size(NN,1)
```

Published with MATLAB® 7.6

APPENDIX VII TCO System Constraints Matlab Script

Contents

- CO subsystem analyser
- Gradient constraints
- local emission constraints
- total cycle emission constraints check
- combine all the different type of constraints


```
function
[c,ceq]=con_sys(x,models,mm_n,res_time,Inputs_Lim,LC,Lim,GC,NN,numb_node,Nu
mofInput)

c=[];ceq=[];

SX=cell(1,10);

loccon=[]; cyclcon=[]; acc_con=[];

Xcounter=[];

for i=1:size(NN,1)

SX{NN(i,1)}=x(:,(size(Xcounter,2)+1):(size(Xcounter,2)+NumofInput(i)));

    Xcounter=[Xcounter
x(:,(size(Xcounter,2)+1):(size(Xcounter,2)+NumofInput(i)))];

end

SX{1,1}=[SX{1,1}(1,:) 69];
```

```
Input argument "NN" is undefined.

Error in ==> con_sys at 7

for i=1:size(NN,1)
```

CO subsystem analyser

```
% parfor i=1:numb_node

%     mode = models{NN(i,1)};

%     LOC_OUT=LC{NN(i,1)};

%     bounds=Inputs_Lim{NN(i,1)};

%     Xsub = SubOpt(SX,mode,bounds,LOC_OUT,GC,NN(i,1)) ;

%     discrepency=abs(Xsub-SX{NN(i,1)}(1,1:6));
```

```
%     acc_con= [acc_con discrepency];

% end

Xsub =
analyser_sys(x,models,res_time,mm_n,Inputs_Lim,LC,GC,NN,numb_node,NumofInpu
t);

acc_con= (Xsub-x).^2;
```

Gradient constraints

```
% acc_con = gradient_check (SX,GC,NN);
```

local emission constraints

```
% parfor i=1:numb_node

%     mode = models{NN(i,1)};

%     Lim_local = LC{NN(i,1)};

%     loccon_em = Local_Emission_check(SX{NN(i,1)},mode,Lim_local);

%     loccon = [loccon loccon_em];

% end
```

total cycle emission constraints check

```
cyclcon = total_emission_check (SX,NN,numb_node,models,mm_n,res_time,Lim);
```

combine all the different type of constraints

```
c=[loccon cyclcon acc_con];

WDEavRCxrA000020000
```

Published with MATLAB® 7.6

APPENDIX VIII TCO Subsystem Matlab Scripts

```
function objsb = obj_sub (x,X0)
```

this function is evaluate the traditional collabarotive optimisation subsystem objective values which is defined by discrepency between system level and sub system level in design space.

```
objsb=sum((x-X0).^2);



% obj=0;

% for i=1:size(GC{N_N,:},2)

%     if ~isempty(GC{N_N,i})

%         obj=obj+sum((x-SX{i}).^2);

%     end

% end
```

```
Input argument "x" is undefined.



Error in ==> obj_sub at 6

objsb=sum((x-X0).^2);
```

```
end
```

```
WDEavRCxrA000020000
```

Published with MATLAB® 7.6

APPENDIX IX TCO Subsystem Constraints Matlab Script

Contents

- local emission constraints
- Gradient constraints


```
function[ con, ceq] = con_sub (x,SX,mode,GC,LOC_OUT,NN)

con=[];ceq=[];acc_con =[];Em_loccon=[];

if NN==1

    x=[x 69];

    SX{NN}=[SX{NN} 69];

end
```

```
Input argument "NN" is undefined.


Error in ==> con_sub at 3

if NN==1
```

local emission constraints

```
Em_loccon = Local_Emission_check(x,mode,LOC_OUT,SX,NN);
```

Gradient constraints

```
SX{NN}=x;

acc_con = gradient_check_sub (SX,GC,NN);

con=[Em_loccon acc_con];

end
```

```
WDEavRCxrA000020000
```

Published with MATLAB® 7.6

APPENDIX X HCO Objective Function Matlab Script (Smoothness 2)

```
function objsb = obj_sub (x,SX,N_N,GC,lb,ub)
```

this function is evaluate the Hybrid Collabarotive Optimisation subsystem objective values which is defined by discrepency between system level and sub system level in design space.

```
% x=x.*(ub-lb)+lb;



SX{1,N_N}=SX{1,N_N};

SX{1,1}=SX{1,1}(1,1:6);

objsb=0;

obj=[];

% w=[2 1 1 2 1 1];

for i=1:size(GC,2)

    if isempty(GC{N_N,i}) || isempty(SX{i})

        continue

    else

        if N_N>i

            diff=x-SX{i};

        elseif N_N<i

            diff=SX{i}-x;

        end

    end

    gradient=[];

    for j=1:size(diff,2)

        if diff(1,j)>0

            gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,2*j)))^2;

%             gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,2*j)));

        elseif diff(1,j)<0
```

```
            gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,(2*j-1))))^2;
%             gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,(2*j-1))));
        else
            gradient(1,end+1)=0;
        end
    end
    obj=[obj gradient];
end
% for i=1:3
%     val=sum(obj);
%     objsb=objsb+val;
%     obj(1,pos)=0;
% end
objsb=sum(obj);
% objsb=max(obj);
```

```
Input argument "N_N" is undefined.

Error in ==> obj_sub at 10
SX{1,N_N}=SX{1,N_N};
```

```
end
```

```
WDEavRCxrA000020000
```

Published with MATLAB® 7.6

APPENDIX XI HCO Objective Function Matlab Script (Smoothness 3)

```
function objsb = obj_sub (x,SX,N_N,GC,lb,ub)
```

this function is evaluate the Hybrid Collabarotive Optimisation subsystem objective values which is defined by discrepency between system level and sub system level in design space.

```
% x=x.*(ub-lb)+lb;



SX{1,N_N}=SX{1,N_N};

SX{1,1}=SX{1,1}(1,1:6);

objsb=0;

obj=[];

% w=[2 1 1 2 1 1];

for i=1:size(GC,2)

    if isempty(GC{N_N,i}) || isempty(SX{i})

        continue

    else

        if N_N>i

            diff=x-SX{i};

        elseif N_N<i

            diff=SX{i}-x;

        end

    end

    gradient=[];

    for j=1:size(diff,2)

        if diff(1,j)>0

            gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,2*j)))^2;

%             gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,2*j)));

        elseif diff(1,j)<0
```

```
            gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,(2*j-1))))^2;
%              gradient(1,end+1)=(diff(1,j)/(GC{N_N,i}(1,(2*j-1))));
        else
            gradient(1,end+1)=0;
        end
    end
    obj=[obj gradient];
end
% for i=1:3
%     val=sum(obj);
%     objsb=objsb+val;
%     obj(1,pos)=0;
% end
%objsb=sum(obj);
 objsb=max(obj);
```

```
Input argument "N_N" is undefined.

Error in ==> obj_sub at 10
SX{1,N_N}=SX{1,N_N};
```

```
end
```

```
WDEavRCxrA000020000
```

Published with MATLAB® 7.6

APPENDIX XII HCO Constraints Function Matlab Script

Contents

- local emission constraints


```
function[ con, ceq] = con_sub (x,SX,mode,GC,LOC_OUT,NN,lb,ub)

con=[];ceq=[];acc_con =[];Em_loccon=[];


% x=x.*(ub-lb)+lb;

if NN==1

    x=[x 69];

    SX{NN}=[SX{NN} 69];

end
```

```
Input argument "NN" is undefined.


Error in ==> con_sub at 5

if NN==1
```

local emission constraints

```
Em_loccon = Local_Emission_check(x,mode,LOC_OUT);

% % %% Gradient constraints

% % SX{NN}=x;

% % acc_con = gradient_check_sub (SX,GC,NN);

con=[Em_loccon acc_con];

end
```

```
WDEavRCxrA000020000
```

Published with MATLAB® 7.6