Development of robust simulation, design and optimization techniques for engineering applications.

Bates, Stuart

Development of Robust Simulation, Design and Optimization Techniques for Engineering Applications

by

Stuart Bates
MEng(Hons) (University of Durham)

Thesis submitted to the University of Wales
in candidature for the degree of
Doctor of Philosophy

School of Engineering, University of Wales, Swansea.

30th September 2003

To my darling Nicola and my Parents

SUMMARY

The present work is concerned with advancing the knowledge base of current CAE tools and to develop new, robust and versatile methods that address industrial design and manufacturing problems through the use of computer simulations and optimization.

The first problem is concerned with slit die design, specifically the shape optimization of a choker bar flow restricting mechanism. Two approaches have been used in solving the problem. The first is deterministic optimization assuming ideal conditions, where the performance of three optimization methods, one gradient-free, one gradient based and one hybrid method, are compared. The second is (stochastic) robust design optimization (RDO) where uncertainty in real conditions is accounted for; in this case the deterministic solution is also a robust design. In developing the RDO procedure two new methods were devised. The first is a multi-objective optimization method, it is more efficient than existing methods. The second is a general method for generating the optimal Latin Hypercube space-filling design of experiments using a permutation genetic algorithm.

The second problem is concerned with the design of an A-pillar for a convertible sports car. A variety of CAE tools are brought together into a single design process. An initial feasible design is generated without the need for prior physical prototyping and testing. The process used to develop the final solution involves topology optimization, shape optimization, response surface approximation, manufacturing simulation and structural simulation.

Thirdly, a general procedure for the development of a CAE system for designing seats for comfort is presented. The system can be applied to the design of seats. It combines robust design and topology optimization. A review of the mechanisms that cause discomfort and some current CAE methods to improve comfort is given.

Finally, an FE-based computer simulation of a cargo during sea transportation is developed to predict the behaviour of the complex multi-body system subjected to a time-dependent, multi-axial loading is described. Load curves to represent the motion of the sea and for simulating the assembly sequences are developed. Using the model, the interactions and the load path sequences for several different loading regimes are identified. They show that there are some major deficiencies in the original design.

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

5th January 2004
Date

Candidate

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. Bibliographies are appended.

5th January 2004
Date

Candidate

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be available to outside organisations.

5th January 2004
Date

Candidate

CONTENTS

ACKNOWLEDGEMENTS

I would like to thank Prof. Peter Bettess for getting me 'hooked' into optimization during my final year project at Durham University. After the project I wanted to get into optimization a little further. Prof. Bettess put me in touch with Prof. Grant Steven who then put me in contact with Dr. Johann Sienz, my PhD supervisor.

I would like to express my sincere thanks to Hans for all the time he has spent helping me, for his constant support and encouragement, and for providing me with an environment in which I feel I have been able to work at my best. I would also like to thank Dr. John Pittman for his encouragement, and for always being willing to help me.

I would like to thank my parents for all their love and understanding, they have been an immense support during this work, and without them it would have been a great deal more difficult. Thanks also to my brothers, Simon and Stephen, for their encouragement and love.

I would like to give my special thanks to Nicola, my rock, for her encouragement, support, patience and understanding, and for her ability to take my mind of my studies and take away all of the stresses. Thank you also to David and Rosemary for being a constant source of support and for being great hosts.

Many thanks also to my friends and colleagues Holger Ettinger, Andreas Rippl and Sava Slijepčević for all of their help on computing issues and for their discussions on other technical issues and to José Fonseca for his help on Chapter 4. I would also like to thank Vassili Toropov, Luis Alvarez, Silvana Afonso da Silva and Andrey Polynkin for their help whenever I needed it.

I would like to thank Royston Jones for giving me invaluable industrial experience at Altair Engineering. Also, thanks to Martin Gambling for his time and support and the rest of the team at Altair for all of their help. Thanks also to Jim Randle and Paul Chapman for providing such an exciting project to work on, and to Masood Turan for help with the manufacturing simulations.

I would also like to thank Tayeb Zeguer, Steve Macmanus and Pam Webster at Jaguar Cars Ltd., for their introduction to, and help with, the seat comfort work.

I would also like to thank the computer administrator Diane Cook and the departmental secretary Margaret Fox for all of their help.

Finally, I would like to thank EPSRC for the financial support.

Abbreviations

ABG	Adaptive Biological Growth
AELH	Audze and Eglais Latin Hypercube DoE
AESO	Additive ESO method
ANN	Artificial Neural Networks
AOF	Aggregate Objective Function
BBD	Box-Behnken DoE
BESO	Bi-directional ESO
BFGS	Broydon-Fletcher-Goldfarb-Shanno
BGSO	Biological Growth Structural Optimisation method
BEM	Boundary Element Method
binGA	binary encoded Genetic Algorithm
CAE	Computer-Aided Engineering
CASS	Computer Automated Seating System
CCD	Central-Composite DoE
CHIM	Convex Hull of Individual Minima
CMR23	Coarse Mesh Result for a minimum member size of 23mm
CONLIN	Convex Linearization
Cosyman	Comfort Optimization SYstem with Mechanical ANalysis
CPU	Central Processing Unit
CT	Computed Tomography
DACE	Design and Analysis of Computer Experiments
DoE	Design Of Experiments
DOT	Design Optimization Tools
E-MOGA	Entropy-Based Multi-Objective Genetic Algorithms
ES	Evolutionary Strategies
EP	Evolutionary Programming
ESO	Evolutionary Structural Optimization
FE	Finite Element
FEM	Finite Element Method
FMR23	Coarse Mesh Result for a minimum member size of 23mm
FMR7	Coarse Mesh Result for a minimum member size of 7mm
FMVSS	Federal Motor Vehicle Safety Standard
FORM	First Order Reliability Methods
FR	Fletcher-Reeves
GA	Genetic Algorithm
Gb	Gigabyte
GESO	Group ESO
GP	Genetic Programming
GRG	Generalised Reduced Gradients method
HS	Hele-Shaw
ICC	Intelligent Cavity Creation
JC	Johnson Controls
LH	Latin Hypercube DoE
MARS	Multi-point Approximations based upon the Response Surface method
Mb	Megabyte

MCS	Monte Carlo Simulation
MESO	Morphing ESO
MMA	Method of Moving Asymptotes
MMFD	Modified Method of Feasible Directions
MMSC	Minimum Member Size Control
MO	Multi-Objective
MOGA	Multi-Objective Genetic Algorithms
MPP	Most Probable Point
MR	Mindlin-Reissner
MRI	Magnetic Resonance Imaging
NBI	Normal Boundary Intersection method
NESO	Nibbling ESO
NNC	Normalized Normal-Constraint method
OLH	Optimal Latin Hypercube DoE
PA	Polynomial Approximations
permGA	permmutation encoded Genetic Algorithm
PFM-PP	Pareto-Front Marching method based on Physical Programming
PP	Physical Programming method
RAMSIS	Computer-based anthropological-mathematical system for passenger simulation
RBDO	Reliability-Based Design Optimization
RCEM	Robust Concept Exploration Method
RD	Robust Design
RDO	Robust Design Optimization
RLH	Random sampling Latin Hypercube DoE
RSM	Response Surface Methodology
SDP	Surrogate Design sPace
SIMP	Solid Isotropic Microstructure with Penalization for intermediate densities
SLP	Sequential Linear Programming
SNR	Signal-to-Noise Ratio
SORM	Second-Order Reliability Methods
SQP	Sequential Quadratic Programming
TSP	Travelling Salesman Problem
VIPS	Vibration Insulation, Pressure distribution, and H-point Simulation
WS	Weighted Sums method

Roman symbols

Scalars

A	cross sectional area
A_i, B_i	lower and upper bounds for design variable of the design variable x_i
A_x	area under normal distribution curve to the left of x
a	microstrcuture cell width
a	lower integral bound
b	microstructure cell height
b	upper integral bound
B	ship width
c	microstructure cell length
C_p	process potential index
C_{pk}	process capability index

$C(0)$, $C(1)$	order of continuity
$dist(i,k)$	distance between adjacent elements i and k
D_h	hydraulic diameter
e_s	elite size
E	Young's modulus of elasticity
ER	evolution rate
f_i	Fitness of an individual i
f_{volume}	volume function
$f_{stress,AC}$	Stress function in member AC
$f_{stress,BC}$	Stress function in member BC
f_1^*, f_2^*	Utopia points
$f(t)$	sea motion function
F	lumbar support force
$F(\underline{x})$	Objective function for design variable set $\underline{x}$
$F_0(\underline{x})$	Objective function
$F(\mathbf{d})$	Objective function for design variable set $\mathbf{d}$
$F(\rho)$	density objective function
$\tilde{F}_j(\underline{x})$	Noise-free approximate response function for design variable set $\underline{x}$
F_e	response parameter
F_i	individual modified objective function value
$F_{\max}$	the maximum modified objective function value
$F_{\max}$	maximum value of the response parameter F_e in a whole structure
$F_{\min}$	minimum modified objective function value
$F_{\min}$	minimum value of the response parameter in a whole structure
$F_{boundary,\min}$	minimum value of the response parameter on the boundary of a structure
g	acceleration to gravity
$g_j(\underline{x})$	typical inequality constraint function
$\bar{\mathrm{g}}_\mathrm{i}$	class function
g_i	value of the i^{th} objective function
$h_k(\underline{x})$	typical equality constraint
H	Planar cavity height
H_e	total gap height
ICR	Insert cavity ratio
IR	inclusion ratio
k	number of points or weightings
K	number of variables
l	number of equality constraints
l	position coordinate along the die exit
$\mathcal{L}$	Lagrangian function
L	die half width
L	Binary string length
L	ship length
L_{pq}	distance between the points p and q

LSL	Lower specification limit
m	number of inequality constraints
m	unladen vehicle mass
M_n , M_{ns}	moments per unit length
n	number of design variables
N	is the maximum design variable value.
n	number of points or weightings
n	number of elements
n	Power law index
p	penalization factor
p	pressure
p_s	population size
P	number of points or levels
P_r	number of points in the same level
$\mathrm{P_r}$	reaction force
$\mathrm{P_{r_{req}}}$	required reaction force
q	normal distributed load per unit area
Q_0	volumetric source term
Q_x , Q_y	local exit flow rates per unit width in x - and y -directions
$\overline{Q}_x$	target exit flow rate
Q_n	transverse shear forces per unit length
res	resolution of catalogued variables
$r_{\min}$	minimum radius of a structural member
RR_i	rejection ratio
S	flow conductivity.
$SL_{closest}$	closest specification limit
t	Time
t_s	tournament size
T	wave period
T	operating temperature
u	displacement of the test rig
U	potential energy
USL	upper specification limit
v_j	slack variable
v_i	elemental volume
$\overline{v}_x , \overline{v}_y$	averaged flow velocities
V	volume
V^t	target volume
W	manifold width
w	z-direction transverse displacement
$w(x, y)$	transverse displacement
x^l , x^u	lower and upper bounds for design variable of the i^{th} design variable
x, y, z	caretsian co-ordinates
X', Y', Z'	local caretsian co-ordinates
y	response value
y	vertical position
Z	response value

Vectors

a_i	tuning parameters
A	lower bounds for design variable set $\underline{x}$
B	upper bounds for design variable set $\underline{x}$
$\mathbf{d}$	vector of design variables
f_i^*	global minima values
F^*	vector containing the individual global minima
$\underline{n}$	normal vector
$\underline{x}$	vector of design variables
$\underline{r}$	search direction
t	distance along a normal

Matrices

D	elasticity matrix
D_s	elasticity matrix of the solid
D^H	updated homogenized elasticity matrix

Greek symbols

Scalars

α	move parameter
γ	cell width
Γ	plate boundary
Γ_S	portion of the plate boundary Γ
ε_b	bending strains
ε_s	shear strain
ε_{p}	maximum plastic strain
E	ergonomics Goodness index
θ	the orientation of a microstructure
λ_j	Lagrangian multiplier
λ_a	shape factor
μ	cell height
μ	mean
$\bar{\mu}^1$, $\bar{\mu}^2$	individual objectives
$\bar{\mu}^{1*}$, $\bar{\mu}^{2*}$	normalized utopia points
μ_{S}	static friction coefficient
μ_{D}	dynamic friction coefficient
ν	Poisson's ratio
ξ	continuous parameter
ρ	density
ρ_i	elemental density
$\rho_{\min}$	minimum density
ρ_s	density of solid
σ	standard deviation
$\sigma_{physical}$	standard deviation of physical data
τ	shear stress

$\phi_{\max}$	maximum angle in radians
ϕ_{rm}	maximum roll angle
ϕ_{pm}	maximum pitch angle
ϕ	melt consistency
$\phi_x(x,y)$	x rotation normal to the mid-plane
$\phi_y(x,y)$	y rotation normal to the mid-plane
Ψ_x, Ψ_y	rotations about the x and y axes
ω	first natural frequency

Vectors

β	Weighting vector
$\dot{\gamma}$	shear rate

Matrices

ϕ	Pay-off matrix

CHAPTER 1

INTRODUCTION

Summary: This chapter gives an introduction to the work carried out in this thesis. The motivation and some results, highlighting key aspects of the research, are given. The overall layout of the thesis is described and can be summarized as follows: Chapters 2, 3 and 4 together represent the first section of the thesis. They provide the descriptions and the developments of the methodologies used in the second section of thesis. The second section, Chapters 5 to 8, consists of several industrial applications involving deterministic and stochastic structural optimization and the use of computer simulations to predict system responses. Chapter 9 concludes the thesis; listing the achievements, conclusions and recommendations for future work.

Engineers make extensive use of computer-aided engineering (CAE) tools to aid in designing and solving engineering problems. The tools are used, e.g. for predicting the response of a structure to environmental factors such as force, temperature and vibration, or for simulating manufacturing processes. This is achieved using computer-based simulations of a system. A simulator is "a device that the operator uses to reproduce or represent, under test conditions, phenomena likely to occur in actual performance."[1.1] The availability of computational simulations allows designers to be able to make rapid assessments of a product or process and generate design alternatives without the need for prior physical prototyping and testing, thereby reducing time and costs; this is the main advantage of CAE.

The type of simulation varies from a simple analytical expression, where the CPU time on a standard PC is in the order of a nanosecond, to a highly complex finite element (FE)-based numerical approximation which may require days or weeks to compute on a multi-processor supercomputer. In some situations, the simple analytical expression may give an exact representation of the behaviour of a system. In most engineering problems exact representations of the system are not possible. There are likely to be multi-body interactions and a model of the complete system is desired over a model that contains only a single component of the complex multi-body system. In any case, the fidelity of the model and the corresponding computational effort required depends on the application, the time constraints and the computational power available.

In engineering design, especially in an industrial environment, it is often critical to be able to rapidly assess the performance of a design, so that design changes can be quickly made to improve the performance. For example, Figure 1.1 shows the performance characteristic (objective function) of a design involving a single parameter (design variable) that can be changed to any value between known limits.

The plot represents a hypothetical 'design space' containing all the possible objective function values for all the design variable values between 0 and 10. In order to achieve the best possible design, the aim of the engineer is to determine the value of the design variable that minimizes the objective function. In this example, it can be clearly seen, that the optimum solution, i.e. minimum value of the objective function is a design variable value of approximately 9.2.

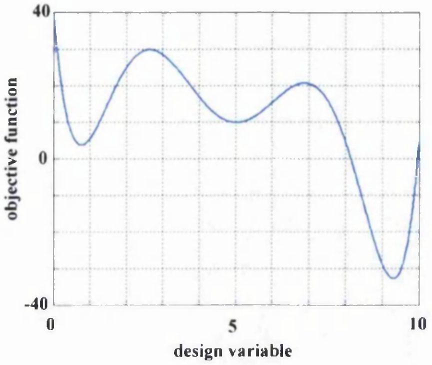


Figure 1.1: Variation of a typical objective function for all values of the design variable between the limits of 0 and 10

In problems with one design variable, as in Figure 1.1, the solution can be obtained by inspection from the complete response curve. For two design variables a typical response surface is depicted in Figure 1.2, and again, the minimum can be determined by inspection.

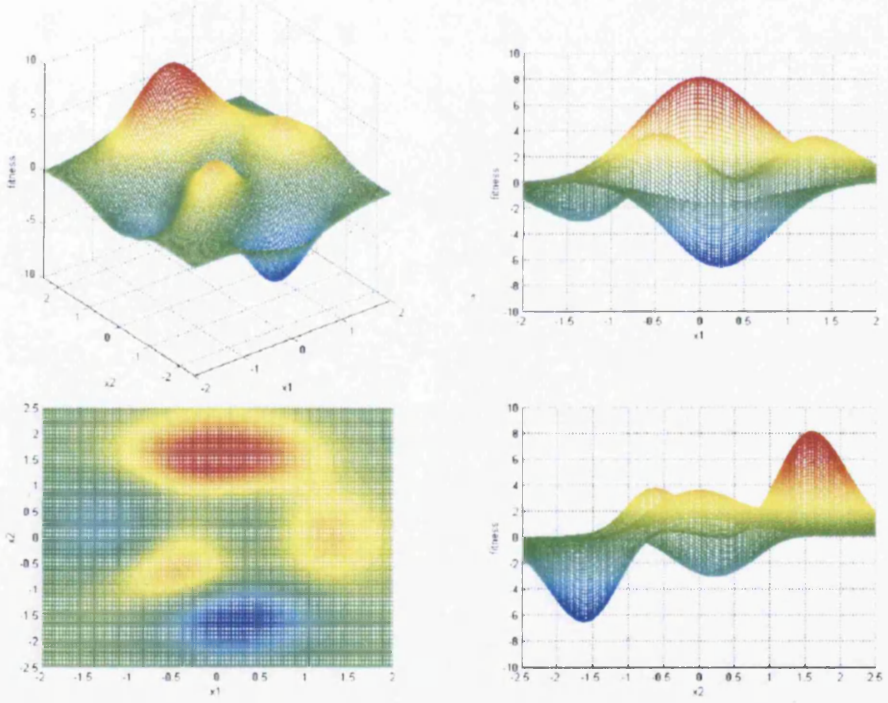


Figure 1.2: A typical two design variable response surface visualized from several positions

However, if there are more than two design variables in a problem, then this visualization method is not so feasible, since more than three-dimensions are required to visualize the design space. Furthermore, the entire design space is not usually available in practice, since simulating all possible combinations of the design variable(s) is not feasible.

Manually assessing the results and making the modifications based on experience is one possible method for achieving improved designs. This ad hoc approach is often cumbersome, expensive and insufficient, since improved designs may be overlooked. Systematic methods for locating the optimum solution come under the heading of 'optimization'.

Optimization is an important CAE tool and it is widely used in engineering design. The idea behind optimization is to find the optimal feasible solution without having to search through all of the possible solutions. Optimization can be applied to search the design space using one or a combination of the optimization methods that are described in Chapter 2.

To solve a problem using optimization requires two items of information: the parameters that can be changed, called the design variables, and a metric of the performance of a solution, the objective function. The optimization process can be split into three parts: models, methods and simulation, according to the 'Three-Columns-Concept' [1.2] in Figure 1.3.

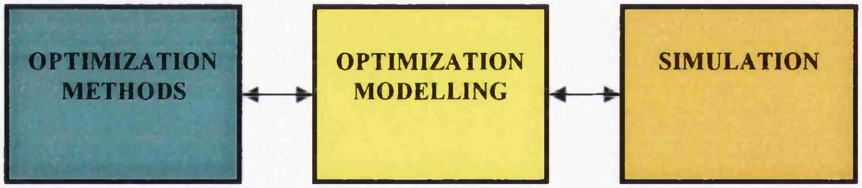


Figure 1.3 Optimization process using the 'three-column concept '[1.2]

To date designers have been reluctant to accept and implement designs that are generated using deterministic optimization, i.e. assuming ideal conditions. This is because the designs are often sensitive to the scatter present in real systems. Until recently, the lack of computational power has meant that it has only been feasible to carry out optimization assuming ideal conditions and using low-fidelity simulations. With the rapidly increasing availability of powerful, inexpensive, multi-processor computers it is possible to develop and run higher-fidelity simulations in parallel and with a reduced CPU time. Overall, this means:

(i) Robust simulations of real systems are feasible and there is a constant need to develop simulations that use the full potential of the CPU available; this is the case with the study described in Chapter 8.

(ii) Optimization considering real variations in a system is also becoming increasingly feasible and designers can generate trustworthy and practicable designs. As such, there is constant need to develop the existing techniques and to introduce new methods to reflect the ever-increasing computational power.

Therefore, there is a need to develop robust simulation, design and optimization techniques; this is the overriding motivation to this work.

Four industrial problems are investigated. To solve the problems requires the use of the techniques encompassed by the three-column concept. The intention is to use, compare and develop existing techniques and where necessary develop new approaches to solve the problems. Therefore, the aim of this thesis is to advance the knowledge base of current computer-aided engineering (CAE) tools and to develop new, robust and versatile methods that address industrial design and manufacturing problems through the use of computer simulations and optimization.

Optimization modelling, described in Chapter 3, is used to parameterize a design problem for use in an optimization process. The parameterization defines the design variables, constraints and boundary conditions. The topics include material distribution methods – topology, shape and size optimization, approximation methods, design of experiments and multi-objective optimization. Two new methods are described in Chapter 3. The first is a method for generating the optimal Latin Hypercube (OLH) space-filling design of experiments (DoE). Its generation is non-trivial and an efficient method for generating the DoE is presented, a typical DoE generated using the method is shown in Figure 1.4, which is a DoE for 120 points and two design variables.

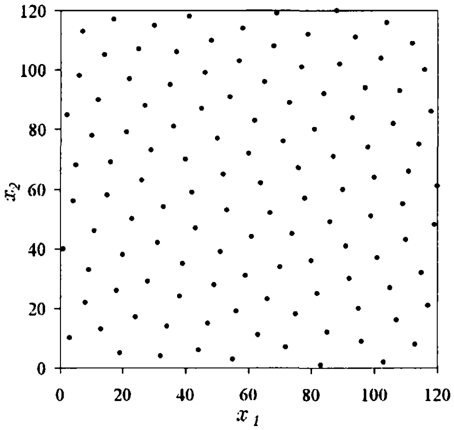


Figure 1.4: OLH DoE for 120 points for two design variables generated using a new technique described in Chapter 3

The other new method described in Chapter 3, is the Pareto-front marching method (PFM-PP). It is a multi-objective optimization method that can be used for generating uniformly distributed Pareto solutions. A typical result, gained using PFM-PP, for a bi-objective problem is illustrated in Figure 1.5, where each point represents a trade-off between the two objectives.

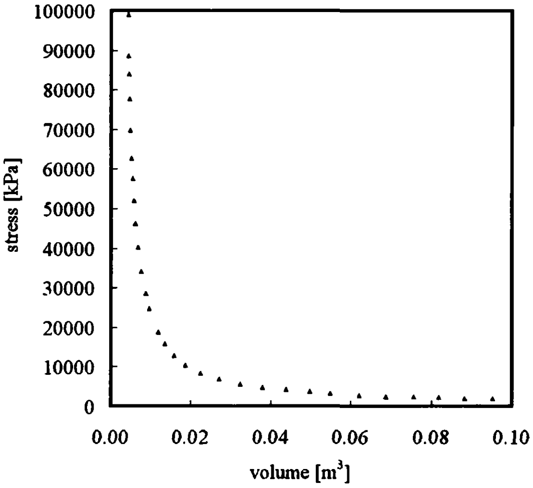


Figure 1.5: Pareto solution set for a bi-objective problem generated using PFM-PP

Optimization methods are the systematic procedures for locating optimum point(s) for a given set of parameters. The performance or the objective function is calculated using either a physical or a virtual (usually) simulation of the problem. Chapter 2 gives a brief overview of the optimization methods, with particular attention paid to the methods used elsewhere in the thesis.

It should be noted, that no matter how complex the model and its parameterization, the search through the design space (either manually or by systematic optimization) can only yield (at best) the optimal solution corresponding to the complexity of the simulation and its parameterization. Therefore, optimization should be seen as a 'virtual' problem solving process, whereby computer simulations can be used to generate a 'good' starting point for solving the real problem. The more complex the model and its parameterization the better the starting point, but this comes at the cost of man hours in developing the model and in CPU time to simulate the model.

Conventional optimization has until recently been applied under the assumption that the problems are deterministic, where the objective is to minimize or maximize a function subject to constraints, and the optimum is calculated under ideal conditions that are difficult to achieve in reality. In reality, uncertain deviations exist in the system properties such as the manufacture, the operating conditions and consequently the performance. As a result of this, carrying out optimization under the assumption that there is no uncertainty doesn't necessarily give the designer a good starting point. It may be the case that the optimal design is ideally optimal but really non-optimal. When using optimization, it can therefore be very important, to account for variation in the properties of the complete system and in the boundary conditions. The methods under this category can be classified as stochastic optimization methods. A stochastic optimization method receiving increasing interest is robust design optimization (RDO). The original idea behind this method dates back to Taguchi [1.3]. In RDO, the aim is not only to minimize or maximize the primary objective function but also to minimize the sensitivity of the solution caused by uncertain deviations. A review of the current methods and applications of RDO is given in Chapter 4 together with the description of the RDO procedure developed in this work for the solution of the industrial problem in Chapter 5.

Chapter 5 is concerned with deterministic and stochastic optimization applied to slit die design. Figure 1.6 gives a schematic of the slit die problem. The aim is to determine the shape of a flow restricting mechanism called a 'choker bar', which restricts the flow of polymer across the die such, that the output is as uniform as possible. The problem, whilst being multi-modal and highly non-linear, requires little computational time to simulate (30 seconds CPU time on a standard PC), it provides a good 'test-bed' for assessing several optimization methods and it is used to develop a multi-objective RDO procedure.

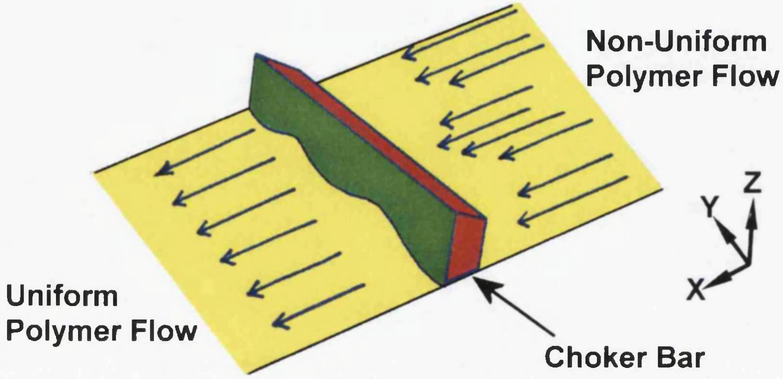


Figure 1.6: Schematic diagram of a choker bar altering the melt flow distribution from a non-uniform distribution to a (ideally) uniform distribution

"The main deficiency has been and to some extent still is that the (optimization) methods have been poorly integrated into the overall design process."[1.4] Chapter 6 addresses this issue and is an industrial example where optimization is the driving force behind the holistic virtual design of an automotive component called an 'A-pillar'. The location of the A-pillar is shown in Figure 1.7.

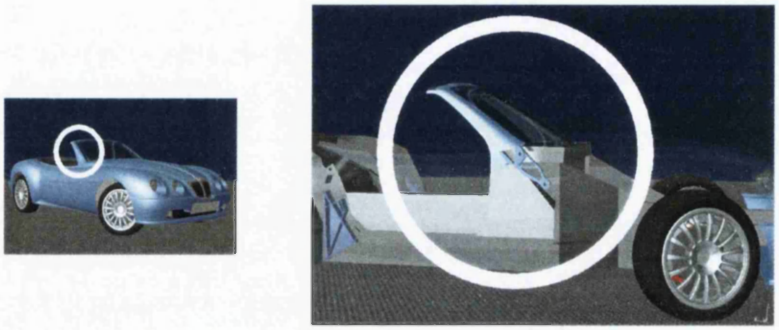


Figure 1.7: Location of the A-pillar under design in Chapter 6

The process used involves topology optimization, shape optimization, response surface approximation, manufacturing simulation and structural simulation. The novelty of this approach is that it brings together a variety of computational design tools into a single design process, thus providing an initial feasible design without the need for prior physical prototyping and testing, and thereby reducing time and costs. The A-pillar is designed under a roof crush loading; Figure 1.8 shows a typical topology result.

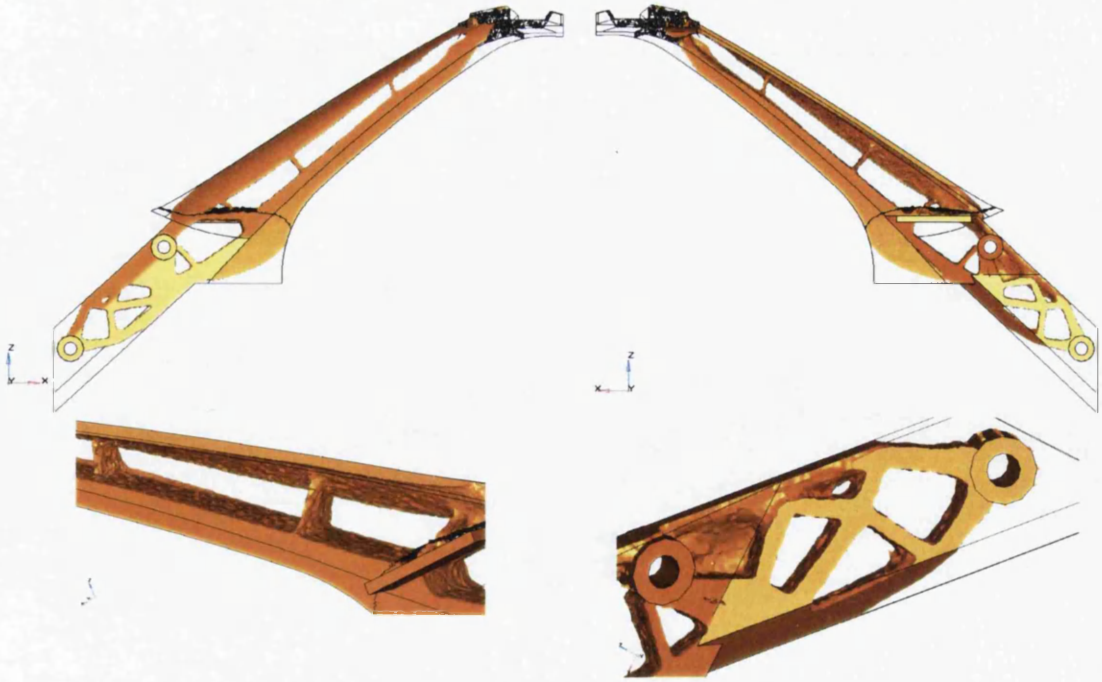


Figure 1.8: topology optimization result from various angles for the A-pillar design in Chapter 6

In general, using CAE helps to improve efficiency and to reduce costs. For example, in the automotive industry where products are constantly being modified and need to remain cost effective, CAE can help to assess many possible solutions in minutes rather than weeks. For instance, Jaguar Cars Ltd. [1.5] currently use questionnaires and physical pressure mapping to identify regions of discomfort in their seats; they require a CAE tool to reduce the number of ‘physical’ design iterations. Chapter 7 is concerned with the development of a CAE tool for designing seats for comfort, using RDO to account for stochastic variation in the shape and the weight of the occupants. An example of initial work carried out into the computer simulation of a dummy sitting on a Jaguar car seat (courtesy of Jaguar Cars Ltd.) can be seen in Figure 1.9, which shows the FE mesh and a typical pressure distribution.

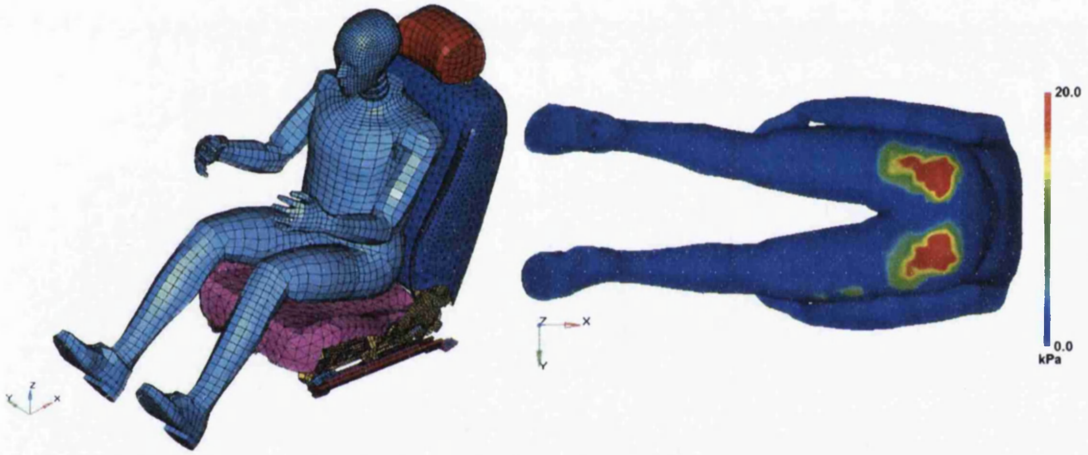


Figure 1.9: Computer model of a dummy sitting on a car seat and the corresponding pressure distribution on the dummy

In Chapter 8, FEs are used to develop a sophisticated simulation model of a cargo inside its transportation packaging. The cargo is a complex multi-body system. Using the simulation, potential problem areas in the packaging used to hold the cargo in place during sea transportation are identified by applying various load cases. For this problem, carrying out physical simulations is infeasible since the cargo is extremely expensive. Furthermore, identifying the interactions and the load path sequence for several different loading regimes would be very difficult, unlike with a computer simulation, where it is relatively simple. Figure 1.10 shows the stages of a typical load time history applied to the FE model. The different forces correspond to the assembly and sea-induced loadings.

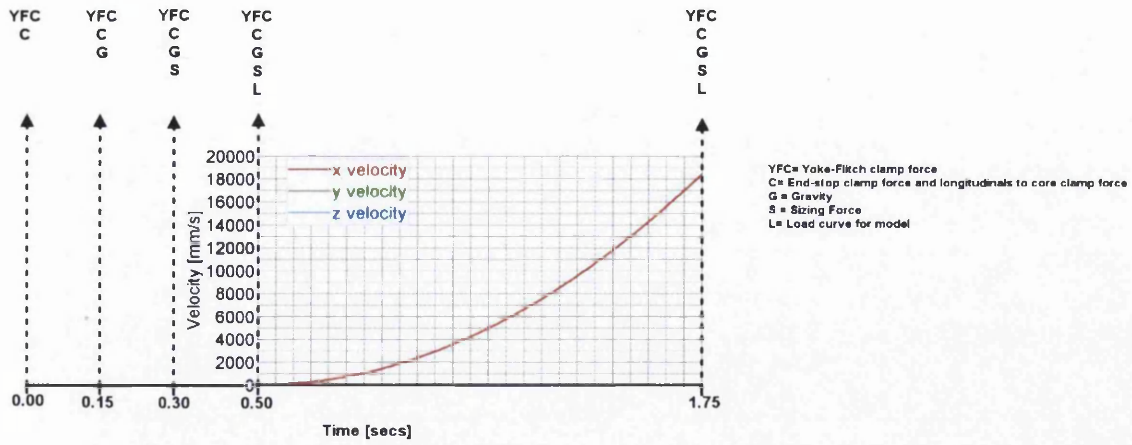


Figure 1.10: The stages of a typical load time history applied to the FE model developed in Chapter 8

Chapter 9 concludes the thesis; listing the achievements, conclusions and recommendations for future work.

1.1 References

[1.1] Merriam-Webster online dictionary. http://www.m-w.com

[1.2] Eschenauer H., Koski J., Osyczka A., Multicriteria Design Optimization, Berlin: Springer-Verlag, 1990.

[1.3] Taguchi G., "Introduction to Quality Engineering: Designing Quality into Products and Processes", Kraus International, New York, 1986.

[1.4] Sienz J., "Integrated structural modelling, adaptive analysis and shape optimization", PhD Thesis, University of Wales Swansea, 1994.

[1.5] Personal communication with Jaguar Cars Ltd.

SECTION 1

METHODOLOGY

CHAPTER 2

OPTIMIZATION METHODS

Summary: In this chapter a brief overview of the optimization methods is given, with particular attention paid to the methods used elsewhere in thesis, i.e. a standard genetic algorithm, the DOT optimization software package and MARS (multi-point approximations based upon the response surface method). The methods that are discussed here are mainly treated as 'black box' optimization tools for the rest of the thesis.

2.1 INTRODUCTION

Section 2.1: Gives an introduction to the chapter.

Section 2.2: Gives an overview of the optimization methods with background information on the various gradient-based, gradient-free and hybrid methods.

Section 2.3: Specific optimization methods that are used in the thesis are described in detail. These are the gradient-based optimizer, DOT, a gradient-free genetic algorithm, and a hybrid method called MARS.

Section 2.4: Gives some concluding remarks.

Section 2.5: Lists the references used in the chapter.

As discussed in Chapter 1, the design space is a hypothetical region, in which all solutions to a parameterized problem exist. It is divided into non-feasible and feasible regions by imposing constraints on the problem. The constraints are normally non-linear functions of the variables; the feasible design space is the region where all of the constraints are satisfied. The optimum solution to a problem is the set of design variables that define a point in the feasible design space corresponding to the minimum (or maximum) value of the objective function. The role of the optimizer, i.e. the method doing the optimization, is to search the design space of a problem looking for the optimal feasible solution.

This chapter is concerned with optimization methods, the first column of the three-column concept depicted in Figure 1.3.

Figure 2.1 (taken from Ref. [2.1]) shows several feasible region shapes for a two-variable design problem.

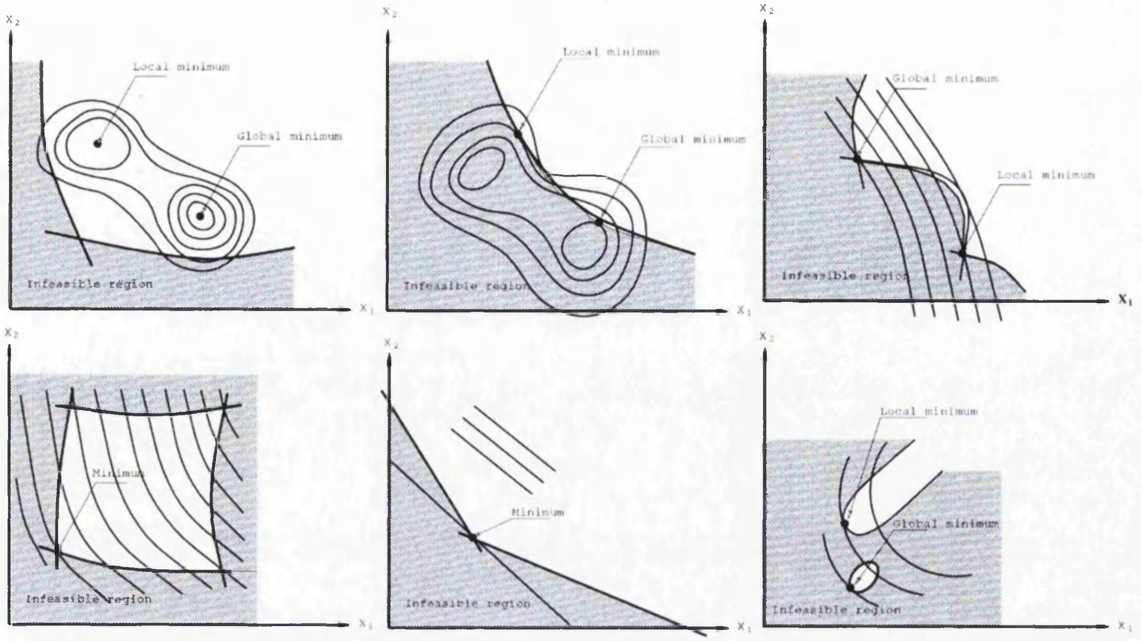


Figure 2.1: Design space configurations: non-linear problem with two minima with non-convex objective function (top left), non-linear problem with two minima on the convex boundary (top centre), non-linear problem with two minima on the non-convex boundary (top right), non-linear problem with one minimum (bottom left), linear problem with one minimum (bottom centre), non-linear problem with two minima and disjoint feasible region [2.1, 2.2]

2.2 Overview of optimization methods

2.2.1 Introduction

The choices of the optimization method and strategy are essential for the successful and efficient solution of the problem. In accordance with Hinton *et al.* [2.1] aspects that are of importance are:

- the type of design variables (discrete or continuous);
- the type of functions (smooth - non-smooth, differentiable, convex - concave);
- constrained or unconstrained design space;
- boundedness of the design space;
- shape of the design space;
- size of the optimization problem (number of design variables, number of functions);
- cost of each simulation;
- multi-modal problem;
- linear or non-linear functions;
- availability of first and second order derivatives;
- local and global optima.

After initialization of the problem the optimizer decides how and where to search the design space. In general, there are two main categories of optimizer [2.1]:

Mathematical programming methods

These methods determine the search direction and the sampling point based on gradient and response information. They normally use continuous design variables and require few function evaluations, but they may not find a global optimum. Some popular methods are sequential quadratic programming (SQP) [2.3, 2.4, 2.5, 2.6], method of moving asymptotes [2.7] (MMA) and generalised reduced gradients (GRG) [2.8].

Evolutionary algorithms

The most common methods in this category are genetic algorithms [2.9], evolutionary programming [2.10], evolution strategies [2.11], particle swarm optimization [2.12] and simulated annealing [2.13]. These methods work mainly using only function values and try to find an optimum by modelling a natural process, such as evolution or the social behaviour of birds flocking as it is the case for particle swarm optimization. These methods can work with discrete variables and are likely to find a global optimum in the presence of many local minima. However, the number of function evaluations can be high.

2.2.2 MATHEMATICAL PROGRAMMING METHODS

"As with all gradient based methods there are two tasks: direction finding or 'where to go' in the design space, and step size selection or 'how far to go'."[2.14] Mathematical programming problems need to be prescribed precisely in mathematical terms. According to Edgar and Himmelblau [2.15] "problem formulation is perhaps the most crucial step in resolving a problem that involves optimization. Problem formulation requires identifying the essential elements of a conceptual or verbal statement of a given application, and organising them into a prescribed mathematical form."

The following classification of the mathematical programming optimization methods is in accordance with Bletzinger [2.16], Sienz [2.17] and Hinton *et al.* [2.1]. It gives a brief overview of the most important features of most of the known algorithms:

Linear programming

These methods are applied problems where the objective function and the constraints are linear. The problems solved have a convex objective function and the linear constraints form a convex set (however, it is difficult to ascertain the convexity of optimization problems in practical engineering). This means that a local optimum will be a global optimum. An advantage of these methods is that the optimal solution of the problem must lie on some constraint or at the intersection of several constraints.

Non-linear programming

Non-linear programming techniques are applied to problems where the objective function is non-linear whereas the constraints are either linear or non-linear. Non-linear problems do not possess the good features of linear programming problems. These techniques are the most popular and widely used methods for structural optimization.

Integer programming

In these methods the variables are not continuous but instead can have integer values only. Integer programming problems are classified as: (i) Mixed-integer – where the objective function depends on a set of integer and a set of continuous variables, (ii) Integer- where there are no continuous variables, and (iii) Binary-integer – where the variables have either a 1 or 0.

Dynamic programming

The idea of dynamic programming is to break down a complex optimization problem into a series of smaller sub-problems that are then solved using one of the other programming techniques.

2.2.2.1 NON-LINEAR PROGRAMMING TECHNIQUES

Non-linear programming techniques are the most popular and widely used methods for structural optimization. They can be divided into methods for the optimization of unconstrained and constrained problems. The methods for unconstrained problems can be split further into techniques for one-dimensional tasks, such as region elimination, uni-dimensional search or polynomial approximation, and multi-dimensional tasks, such as direct search methods, indirect methods of first and second order or secant methods. The one-dimensional methods are quite often an important part of multi-dimensional methods as the convergence of the latter can be ensured by successful application of one-dimensional line searches. The methods for the optimization of constrained non-linear problems can be divided into four major categories [2.16]: primary methods, penalty and barrier methods, dual methods and Lagrange multiplier methods. [2.1]

2.2.2.1.1 PRIMARY METHODS

These methods work in the feasible space of the problem and require that the initial design is feasible. By always remaining in the feasible region the solutions remain practicable even if the optimization process terminates prematurely. The methods are gradient-based methods that use a line search to ensure that the constraints are satisfied. The methods in this category include the generalized reduced gradient method (GrG2) by Lasdon *et al.* [2.8], and the method of feasible directions, e.g. by Vanderplaats and Moses [2.18].

2.2.2.1.2 PENALTY AND BARRIER METHODS

The idea of these methods is to transform the problem into a problem in which a single unconstrained penalty or augmented function is minimized. The original constrained problem is solved by a systematic series of minimizations of unconstrained functions of the same form but with different parameters. The disadvantage of these methods is that the exact solution cannot be reached. The exact solution is approached by the penalty method from the infeasible space and by the barrier method from within the feasible region.

2.2.2.1.3 DUAL METHODS

These methods solve the optimization indirectly by formulating the dual problem, the unknowns of which are the Lagrange multipliers. The number of the multipliers is equal to the number of side constraints. The values for the design variables can be found by back-inserting the solutions for the Lagrange multipliers. The problem can be split into two parts of which the first one is unconstrained. The second problem has simple limits for the variables in the form of inequality constraints. If there are only equality constraints the second problem is unconstrained as well. These parts can be solved applying the methods for unconstrained optimization. Known implementations of this group are the CONLIN algorithm (convex linearization) by Fleury [2.19] or in more general form the MMA algorithm by Svanberg [2.7]. Both methods combine successfully dual methods, approximation techniques and separable problems. The results are efficient algorithms, which can be employed for the solution of large structural optimization problems and are especially useful for problems where the number of design variables is large, e.g. using homogenization methods in topology optimization (discussed in Chapter 3) where typically the number of variables is at least equivalent to the number of designable FEs.

2.2.2.1.4 LAGRANGE MULTIPLIER METHOD

These methods aim to find a direct solution for the necessary condition of optimality, i.e. the Kuhn-Tucker conditions (Section 2.2.2.3). They combine the features of the primary and the dual methods. The resulting equations are solved iteratively either with Newton or Quasi-Newton methods. Locally, the objective function is approximated by a quadratic function, and the constraints by linear functions. The sub-problems, which are quadratic, are solved by means of quadratic programming methods. Therefore these methods are also known as sequential, successive, or recursive quadratic programming methods. Sequential quadratic programming (SQP) methods belong to the most powerful non-linear programming algorithms used for solving differentiable, non-linear programming problems. Stoer [2.20] describes the theoretical background. Sequential linear programming (SLP) is another popular Lagrange multiplier methods where the sub-problem is linear and is solved by a simplex algorithm.

2.2.2.2 OPTIMIZATION FORMULATION

In this section, the basic formulation of optimization problems is presented as in the work of Hinton *et al.* [2.1]. A typical optimization problem can be defined by the following expressions:

$$\begin{array}{rlr} \text{minimize:} & F(\underline{x}) & \\ \text{Subject to:} & g_j(\underline{x}) \le 0; & j = 1,\ldots,m \\ & h_k(\underline{x}) = 0; & k = 1,\ldots,l \\ & x_i^l \le x_i \le x_i^u; & i = 1,\ldots,n \end{array} \tag{2.1}$$

where $\underline{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of the design variables, $F(\underline{x})$ is the objective function, $g_j(\underline{x})$ is a typical inequality constraint function, $h_k(\underline{x})$ is a typical equality constraint function and x_i^l and x_i^u are the upper and lower bounds on the i^{th} design variable, respectively. In addition, m is the number of inequality constraints, l is the number of equality constraints, and n is the number of design variables. The formulation can be simplified by using the fact that an equality constraint can be replaced by two inequality constraints and Equation 2.1 can be rewritten as

$$\begin{aligned} \text{minimize: } & F(\underline{x}) & & \\ \text{Subject to: } & g_j(\underline{x}) \le 0; & & j = 1,\dots,(m+2l) \\ & x_i^l \le x_i \le x_i^u; & & i = 1,\dots,n \end{aligned} \tag{2.2}$$

2.2.2.3 KUHN-TUCKER OPTIMALITY CONDITION

In order to determine if a solution is optimal (at least locally), then firstly a design must pass the necessary conditions which are determined using the Lagrangian multiplier method and then the Kuhn-Tucker conditions. This section describes these optimality conditions in accordance with Hinton *et al.* [2.1] and Lee [2.21].

Based on the Lagrangian multiplier method, the objective function can be augmented as

$$\mathcal{L}(\underline{x},\underline{v},\underline{\lambda}) = F(\underline{x}) + \sum_{j=1}^{m+2l} \lambda_j (v_j^2 + g_j(\underline{x})) \tag{2.3}$$

where v_j is a slack variable and λ_j is a Lagrangian multiplier. Hence, the necessary condition for a candidate solution to provide an extremum of the Lagrangian function $\mathcal{L}$, which may also be considered as the necessary condition for Equation 2.1 to be satisfied, can be written as

$$\frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial F(\underline{x})}{\partial x_i} + \sum_{j=1}^{m+2l} \lambda_j \frac{\partial g_j(\underline{x})}{\partial x_i} = 0 \qquad i = 1,\dots,n \tag{2.4a}$$

$$\frac{\partial \mathcal{L}}{\partial v_i} = 2\lambda_j v_j = 0 \qquad j = 1,\dots,(m+2l) \tag{2.4b}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_i} = (v_j^2 + g_j) = 0 \qquad j = 1,\dots,(m+2l) \tag{2.4c}$$

where Equation 2.4b determines whether the constraints are active. If the constraint is active then $g_j = 0$, while if it is inactive then $\lambda_j = 0$. Also, Equation 2.4c ensures that the inequality $g_j \le 0$ for all j. The design $\underline{x}$ may be considered to be a minimum if it is feasible and if a set of Lagrangian multipliers λ_j exists such that

$$\nabla F(\underline{x}) + \sum_{j=1}^{m_a} \overline{\lambda}_j \nabla g_j(\underline{x}) = 0 \tag{2.5}$$

where m_a is the number of active constraints and $\bar{\lambda}_j$ belongs to the set of λ_j. In the case of a two dimensional design space, the direction of steepest descent $-\nabla F(\underline{x})$ can be written as a linear combination of the constraint normals

$$-\nabla F(\underline{x}) = \bar{\lambda}_1 \nabla g_1(\underline{x}) + \bar{\lambda}_2 \nabla g_2(\underline{x}) \tag{2.6}$$

and it may be illustrated as shown in Figure 2.2 (left). Even though the design $\underline{x}$ might satisfy the necessary conditions imposed by the Lagrangian multiplier method, the design $\underline{x}$ may not be a local minimum. Therefore, in order to ensure the solution of (2.2) is a local optimum, the Kuhn-Tucker conditions [2.22]

$$\frac{\partial F(\underline{x})}{\partial x_i} + \sum_{j=1}^{m_a} \lambda_j \frac{\partial g_j(\underline{x})}{\partial x_i} = 0 \qquad i = 1,\ldots,n \tag{2.7a}$$

$$\lambda_j g_j(\underline{x}) = 0 \qquad j = 1,\ldots,m_a \tag{2.7b}$$

$$\lambda_j = 0 \qquad j = 1,\ldots,m_a \tag{2.7c}$$

must be used where (2.7c) ensures that the design is a local optimum if $\nabla F(\underline{x})$ lies in the sector defined by ∇g_1 and ∇g_2 in the case of a two dimensional design space with two active constraints [2.23] (see Figure 2.2, right).

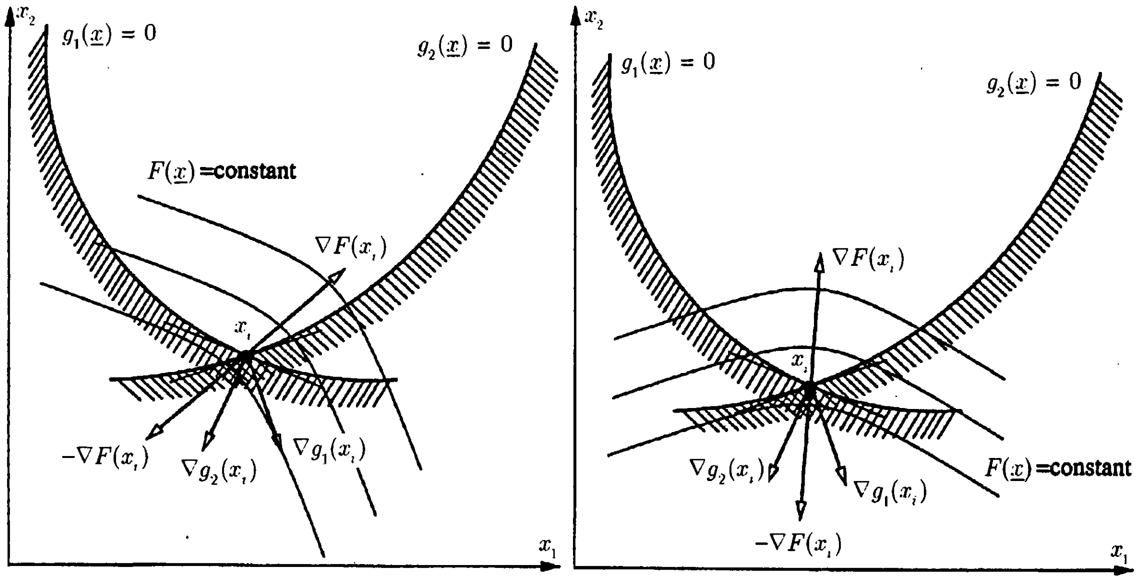


Figure 2.2: Mathematical programming: satisfying the necessary condition (left) and the Kuhn-Tucker conditions (right) [2.21, 2.1]

The geometric interpretation of these conditions given by [2.1] means that – since the sum of the gradient of the objective function and the linear combination of the gradients of the active constraints at the optimum is equal to zero [2.24] – the negative gradient vector of the objective function is inside a cone built by the positive gradient vectors of the active constraints.

2.2.2.4 GENERAL OPTIMIZATION STRATEGY

The optimization problem can be considered as an iterative procedure to find a better solution. The most common form of the iterative optimization procedure may be expressed as

$$\underline{x}^{q} = \underline{x}^{q-1} + \alpha \underline{r}^{q} \tag{2.8}$$

where α is the optimum move parameter, $\underline{r}$ is the search direction and the superscript q denotes the iteration number. In order to utilize Equation 2.8, two items are required: (a) the determination of the search direction $\underline{r}$ and (b) the interpolation of the move parameter α which will minimize (or maximize) the objective function $F(\underline{x})$ in the direction $\underline{r}$. The detailed descriptions of the strategies for calculating the search direction $\underline{r}$ and the interpolation of the move parameter α can be found in Ref. [2.25] and Ref. [2.23] respectively, and also in Ref. [2.1].

2.2.2.5 CONVERGENCE CRITERIA

Several convergence criteria may be adopted. They are usually checked after every iteration and are typically as follows (as in the work of Hinton *et al.* [2.1]):

1. **Maximum number of iterations**

The optimization will continue up until the iteration number provided by the user. This criterion can avoid unnecessary computation during optimization iterations.

2. **No feasible region**

If a feasible solution is not found in a specified number of iterations, then the iteration process is terminated.

3. **Norm of the objective function**

If the Euclidian norm of the objective function $\left|F^{q} - F^{q-1}\right| / \left|F^{q-1}\right|$ between two subsequent iterations is less than a specific value, then the iteration process is terminated.

4. **Change of the design variables**

If the norm of $\left\|\underline{x}^{q} - \underline{x}^{q-1}\right\| / \left\|\underline{x}^{q-1}\right\|$ between two subsequent iterations is less than a specific value, then the iteration process is terminated.

5. **Absolute change of the value of the objective function**

If the absolute change of the objective function $|F^{q} - F^{q-1}|$ is less than a specified value, then the iteration process is terminated.

6. **Kuhn-Tucker conditions**

In the constraint minimization (or maximization) problem, the gradient of the augmented Lagrangian $\nabla F(\underline{x}, \lambda)$ at the optimum solution should vanish. If so, the iteration process will be terminated.

2.2.3 EVOLUTIONARY ALGORITHMS

In this section a brief overview of some evolutionary methods is given. These methods exploit a set of potential solutions, named a *population*, and "detect the optimal problem solution through co-operation and competition among the individuals of the population"[2.26].

2.2.3.1 GENETIC ALGORITHM

GAs were originally proposed by Holland [2.27], they are search procedures based on natural selection and survival of the fittest. In them, organisms that are better adapted to their living environment are more likely to survive and pass their genes on to the next generation than less well-adapted organisms, because individuals carrying them get more chances to breed. Hence, characteristics that improve the fitness of the organism are more likely to be passed on to the next generation. Unlike mathematical programming algorithms, the GA search method does not require information about gradients or sensitivities of the objective function and can cope with both continuous and discrete data if appropriately coded. GAs are best suited for unconstrained optimization problems. But, by using a penalty-based transformation method, constrained problems can be converted into unconstrained ones. See Goldberg [2.9], Rajeev and Krishnamoorthy [2.28], Ghasemi [2.29], Ghasemi and Hinton [2.30], for examples of some penalty functions.

The analogy between the 'survival of the fittest' principle and the GA is as follows. In nature, a set of organisms, each with its own unique characteristics, makes up a population. In the GA a set of solutions to a problem, i.e. a 'population' of designs, are created. Each design variable is a *gene* and each design has a unique combination of the design variables called a *chromosome*. The best designs, i.e. the *fittest* designs, satisfying the objectives and the constraints of a problem are given more chance to survive and pass on their characteristics. “By keeping many solutions in the pool during the search process, rather than converging on a single point early in the process, we reduce the risk of converging to a local minimum” [2.31]. Evolution takes place during reproduction and is driven by mechanisms known as crossover and mutation of the parents’ chromosomes leading to new and mainly fitter children, in this case new designs. Usually, two children are generated by ‘mating’ two parents together. Section 2.3.2 gives further details on a typical GA including encoding methods, parent selection, reproduction operators, control parameters, convergence criteria and rebirthing.

2.2.3.2 EVOLUTIONARY PROGRAMMING AND EVOLUTIONARY STRATEGIES

Bäck *et al.* [2.32] give a comprehensive comparison between evolutionary strategies (ES) and evolutionary programming (EP). EP was initially developed by Fogel L.B. *et al.* [2.33] and later generalized for numerical optimization by Fogel D.B. [2.10]. The idea of EP is similar to GAs, but it also differs in a number of ways. (i) Each member of the population generates a single child by *mutation* only; (ii) EP uses the design variable values directly and therefore an encoding method is not required; (iii) the mutation operation makes minor mutations of the offspring highly likely and substantial mutations as increasingly unlikely; (iv) the severity of mutations is often reduced as the global optimum is approached. ES were developed by Bäck *et al.* [2.11] and are similar to EP although the two approaches were developed independently. ES typically uses deterministic selection in which the worst individuals are removed from the population whereas with EP the selection of parents is done in a random manner typically using tournament selection (discussed in Section 2.3.2.2.2).

2.2.3.3 PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization (PSO) was originally proposed as a simulation of social behaviour, and then introduced as an optimization method by Eberhart and Kennedy [2.12]. The underlying principle of PSO (as explained by Schutte *et al.* [2.34] for instance) is that there are number of particles which co-ordinate searches in the design space by communicating the locations of promising regions. At a time step, each particle in the swarm occupies a distinct point in the design space, and has a pseudo velocity and inertia. At every iteration, the fitnesses of the particles are evaluated and their velocities are adjusted according to the most promising location found by the swarm and themselves.

Parsopoulos and Vrahatis [2.26] discuss recent approaches to optimization using PSO. Venter *et al.* [2.35] carry out robust design optimization comparing PSO to GAs. For the first time a parallel version of PSO has been implemented by Schutte *et al.* [2.34]. Other applications are Wilke and Groenwold [2.36], Yu and Zhaoying [2.37].

2.3 SPECIFIC OPTIMIZATION METHODS

In this section a description of the optimization methods used elsewhere in the work is given.

2.3.1 MATHEMATICAL PROGRAMMING METHOD – DOT

DOT is a commercial general-purpose numerical optimization software package. Vanderplaats [2.3] gives full details on the software, it is considered here as 'black box'. Some of the methodologies implemented in the software are briefly explained in Section 2.2.2. The version of the DOT software used during this work contains the following optimization methods:

- For constrained optimization: Modified Method of Feasible Directions (MMFD), Sequential Linear Programming (SLP) with adjustable move limits and Sequential Quadratic Programming (SQP).
- For unconstrained optimization (allowing for lower and upper bounds on the design variables): Broydon-Fletcher-Goldfarb-Shanno (BFGS) algorithm and Fletcher-Reeves (FR) algorithm.

2.3.2 EVOLUTIONARY ALGORITHM – GENETIC ALGORITHM

Section 2.2.3.1 gives an introduction to GAs.

2.3.2.1 ENCODING

The ‘encoding’ of the design variables into chromosomes is the first step when solving a problem using a GA. There are a number of different methods including binary, gray, permutation, value, and tree encoding; the type used depends on the problem. The design variables in a GA are a discrete catalogue of values represented by an encoding. Pseudo-continuous variables are used, since the continuous variables can only be approximated using a catalogue of discrete values. The difference between the catalogued values is termed the *resolution*. In general, if L bits are used, the resolution res can be obtained from the following expression

$$res = \frac{x^u - x^l}{2^L - 1} \tag{2.9}$$

where x^l and x^u are the upper and lower bounds of the design variable. For discrete problems, res is fixed according to the number of possible discrete values. For continuous problems, i.e. problems where the number of possible solutions is infinite, res is changed according to the desired accuracy. If $res = 0$, i.e. there are an infinite number of values in the catalogue, then the discrete approximation will exactly model a continuous set of design variables.

2.3.3.1.1 BINARY ENCODING

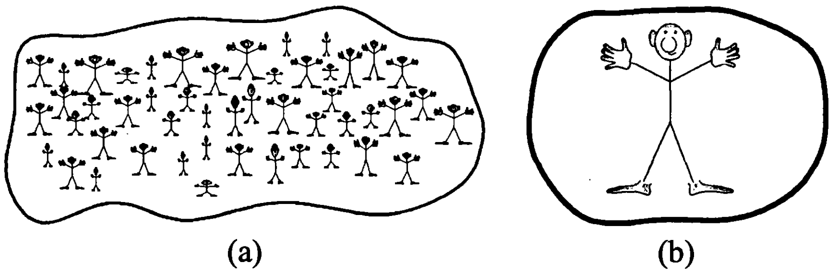
Binary encoding was the first type to be used for GAs due to its relative simplicity. In binary encoding, every chromosome is a string of bits - 0 or 1. Binary encoding gives many possible chromosomes even with a small number of genes. For example [2.1], Table 2.1 shows a typical discrete set of values of design variables for an optimization problem; if the design variable has an integer value of 20, then its encoded value in binary form is 10100, which corresponds in Table 2.1 to a real design variable value of 7.192.

Value	Encoding variables												
Cat. Value	0	1	...	8	9	...	15	...	20	21	...	30	31
Encoded	00000	00001	...	01000	01001	...	01111	...	10100	10110	...	11110	11111
Real value	0.001	0.100	...	1.333	1.488	...	3.565	...	7.192	8.525	...	33.700	37.200

Table 2.1: Typical table of values with binary encoding [2.1]

An example of how a bit string can characterise an individual in a population is given in Example 2.1.

Example 2.1 – An initial, randomly chosen, population of 50 people is shown in Figure 2.3(a). Each individual in the population has five genes that can have two values: nose (big or small), mouth (big or small), ears (big or small), legs (short or long), and feet (big or small).



Big Ears	1
Big Nose	1
Big Hands	1
Big Feet	1
Long Legs	1

(c)

Figure 2.3 (a) A randomly chosen population of 50 people; (b) An individual chosen at random and (c) its chromosome

Figure 2.3(b) shows a member of the population chosen at random and has a chromosome of Big Ears, Big Nose, Big Hands, Big Feet, and Long Legs. The chromosome can be represented by combining the gene bits into a binary string of 0s and 1s as can be seen in Figure 2.3(c). Because there are only two possibilities for each gene, the gene is assigned either a '1' or a '0'. If it were possible to have a small, medium or large nose then the 'nose' gene would require a two-bit string consisting of values such as 01,10 or 11 for small, medium, or large respectively.

This encoding is often not suitable for some problems and corrections must be made after crossover and/or mutation. For example, in the travelling salesman problem (discussed later in Section 3.4.2.4) the repetition of a catalogued value is prohibited. Encoding the problem using binary encoding would make it highly likely that an infeasible design be formed when the traditional crossover and mutation operators are applied. So, penalization of the solutions is required to filter out infeasible designs and computation time is wasted on evaluating infeasible solutions.

2.3.3.1.2 GRAY ENCODING

Gray encoding uses binary numbers and only differs from binary encoding in the order of the binary sequence. Gray encoding sequences are organized such that adjacent integers have representations that differ in only one bit position. Langley [2.38] compares in detail gray encoding to binary encoding.

2.3.3.1.3 Permutation Encoding

Permutation is "the rearrangement of existent elements, or the process of changing the lineal order of an ordered set of objects"[2.39]. Permutation encoding can be used in ordering problems, such as the travelling salesman problem or in developing DoE (Section 3.4.2.4). In permutation encoding, every chromosome is a string of numbers that represent a position in a sequence. For example a travelling salesman has to visit five cities, if each city can is represented by an integer value, then the chromosome representing the route, i.e. a member of the population, consists of a sequence of non-repeated integer values (1-5 in this case).

By using a permutation encoding instead of binary encoding the problem of infeasible designs being generated (discussed in Section 2.3.3.1.1) can be overcome; especially as it is possible to model the mutation and crossover operators (described in Section 3.4.2.4) such, that infeasible designs cannot be produced. A comparison between permutation encoding and binary encoding is given in Section 3.4.2.4

2.3.3.1.4 Value encoding

Direct value encoding can be used in problems where some more complicated values are used. In the value encoding, every chromosome is a sequence of some values. Values can be anything connected to the problem, such as real numbers, characters and directions.

Chromosome	Encoding
1	1.354 3.234 7.564 2.345 35.435 1.246 7.655
2	AQYUGKMVXUWOJXFAKSDO
3	up, down, left, up, down, right

Table 2.2: Example of value-encoded chromosomes

2.3.3.1.5 Tree encoding

Tree encoding is used in genetic programming for evolving programs or expressions. In this encoding every chromosome is a tree of some objects, such as functions or commands in programming language. Genetic programming is discussed further in Section 3.3.2.1.5.

2.3.2.2 SELECTION OF PARENTS

In order for a population of designs to evolve, a population of parents are ranked according to their fitness and placed in a 'mating pool'. For unconstrained design problems, the fitness of a design can be defined as the objective function (for maximization problems) or a constant minus the objective function (for minimization problems). For constrained problems the fitness may be penalized if the constraint is violated. Two parents normally reproduce to form two children. So, if there are p_s parents in a population then, $p_s / 2$ matings are required to produce p_s children for the next population. If elitist selection (described below) is chosen, then the number of matings will reduce by the number of elite members chosen. An option, sometimes used at this stage is to kill off the weakest members of the population by either specifying a 'weakest' percentage or by eliminating designs below the average fitness.

The *Selection* operator selects members of the population for reproduction. It does this by mimicking biology and giving fitter individuals (or designs) a higher chance of reproducing, hence the fitness of an organism is defined by the likelihood that the organism will survive to reproduce. There are various methods for selecting the fittest designs for reproduction. The most common methods are the 'Roulette-Wheel' analogy first considered by Goldberg [2.9] and the tournament selection [2.9].

2.3.2.2.1 ROULETTE WHEEL SELECTION

An analogy often considered is that of a roulette wheel with different sized slots. Each slot is associated with a parent and the greater the fitness of the parent the greater the slot size. The parents are randomly selected, which implies in this analogy allowing a roulette ball to fall into a slot in a spinning roulette wheel. There is a higher likelihood that the ball will land in the larger slots more often and, by analogy, parents with greater fitness will be chosen to mate more frequently.

2.3.2.2.2 TOURNAMENT SELECTION

When using tournament selection, a tournament size t_s is defined *a priori* (usually in an input data file). A sub-population of size t_s is randomly chosen, and the individual with the highest fitness wins the tournament and becomes a mating parent. Once two parents are chosen then the reproductive operators (discussed in the next section) are applied to produce two children, which are moved into the next generation. All of the sub-population (including the tournament winner) is then placed back into the old population and the process is repeated until the new population is full. "Since the size of the sub-population ranges from 5 to 20% all but the very worst designs are given some chance of passing on to the next generation. Multiple pairings of fit individuals can and do occur. Tournament selection can give faster convergence to an optimum solution than other selection methods such as roulette wheel."[2.38]

2.3.2.2.3 ELITIST SELECTION

Elitist selection is applied to save the best e_s percentage of a population. It does this by 'fast-tracking' the fittest e_s% of designs directly into the next generation and replacing the lowest e_s% of the population. "This selection process therefore provides a safety net, in case the genetic operations do not produce desirable individuals, and ensures the survival of the fittest designs from the previous generation to the next."[2.38]

2.3.2.3 REPRODUCTION- CROSSOVER AND MUTATION

In nature, new generations are created by the swapping of genetic information between parents to produce offspring. The swapping of genetic information is carried out by reproduction and is called *crossover.* Errors sometimes occur during crossover and these errors are called *mutations* e.g., a '1' is passed over instead of a '0' or big ears are produced instead of small ears in Example 2.1. Mutation occurs in nature by randomly changing a part of the genetic information; it is an important concept of a GA as it acts like a safety net preventing the premature loss of useful genetic information and premature convergence. Example 2.2 continues from Example 2.1 to show how the genetic operators, crossover and mutation, work.

Example 2.2 – Figure 2.4 shows an individual chosen by selection (Section 2.3.2.2) from the population shown in Figure 2.3(a). It has small ears, small nose, small hands, small feet, and short legs, i.e. 0,0,0,0,0 in binary form.

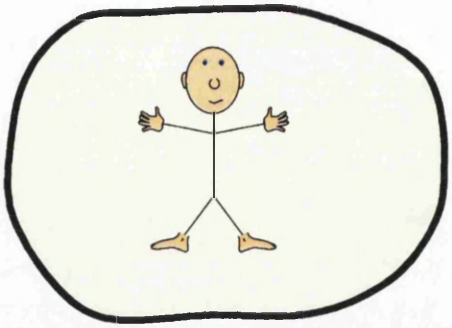


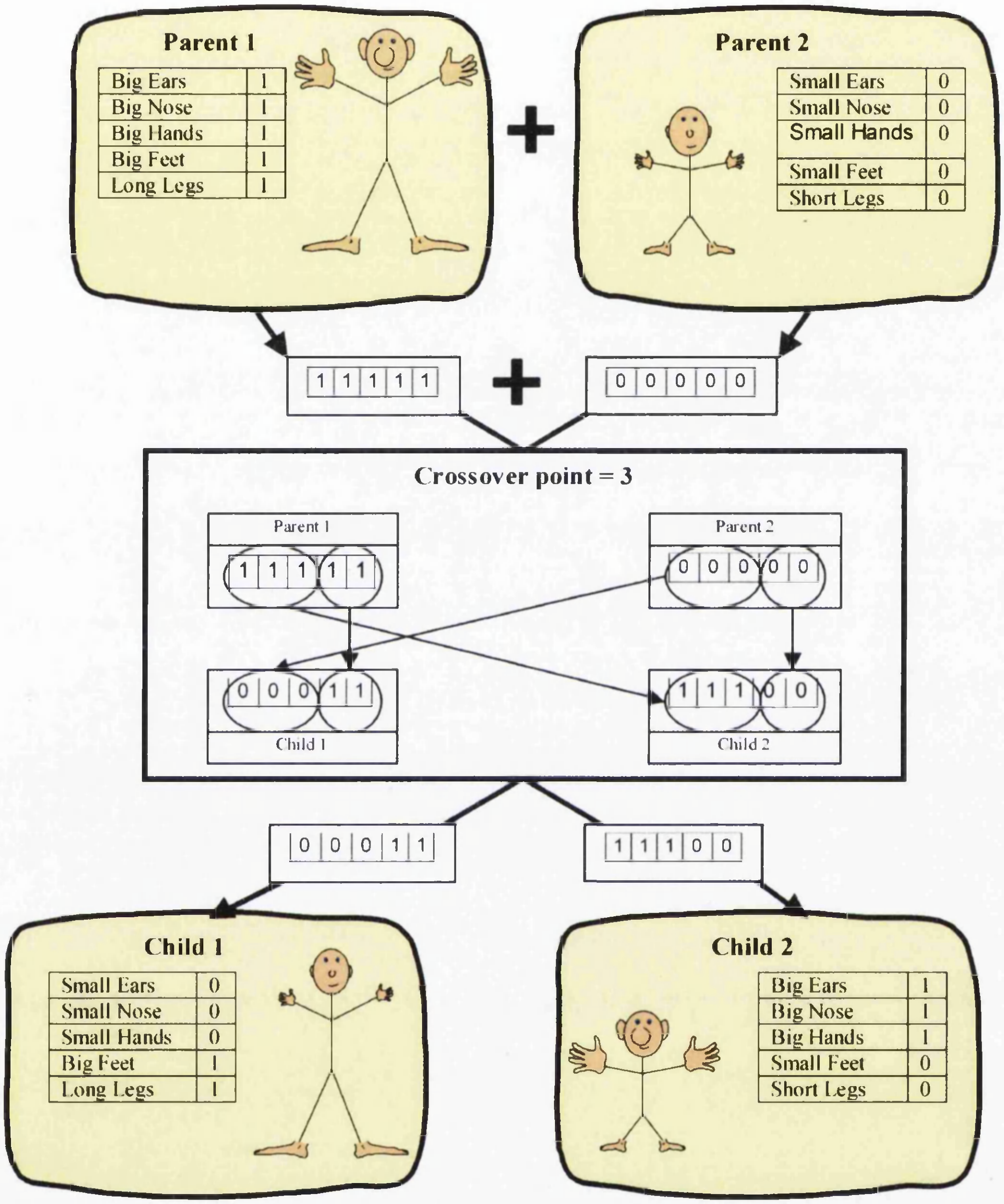
Figure 2.4: A second member of the population in Figure 2.3(a) chosen at random

A tournament selection might result in the two individuals in Figures 2.4 and 2.3(b), to be chosen for reproduction, resulting in two children being produced. The children can have any combination of the parents' characteristics for example, big ears, big nose, small hands, small feet, and short legs. The reproduction process is implemented in the GA by the so-called crossover and mutation operators.

The simplest crossover method is *one-point* crossover. Figure 2.5 shows a one-point crossover between the chosen individuals. Genetic information in the GA is held in a binary string, in this case of length 5. One-point crossover swaps genetic information between the parents at a single crossover point, for this example point 'three' is randomly chosen. The corresponding children are then produced. This can be seen in Figure 2.5 where child 1 is produced by parent 1 swapping the first three bits of its chromosome with the first three bits of parent 2's chromosome. So, child 1 has the same first three characteristics as parent 2 and the same last two characteristics as parent 1 (vice-versa for child 2).

This process can be extended to multi-point crossover, whereby genetic information is swapped by repeated one-point crossovers.

The next operation of a simple GA is mutation, which plays a secondary role in the operation of the GA. Mutation is needed because, even though reproduction and crossover effectively search and recombine existing chromosomes, it additionally allows new genetic patterns to be formed improving the search method. In artificial genetic systems the mutation operator occasionally protects some useful genetic material against loss. Mutation is implemented in the GA by randomly altering a point in the binary string from 0 to 1 or 1 to 0. Figure 2.6 shows what the genetic makeup of child 1 in Figure 2.5 would become if introducing a mutation at a randomly chosen point, point 2.



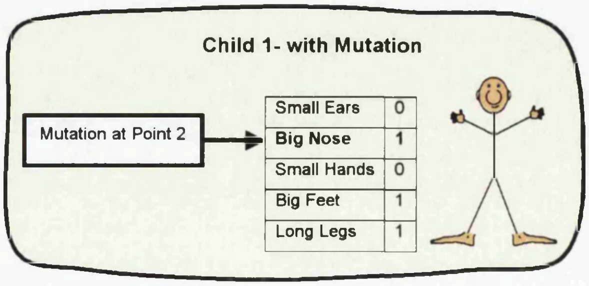
Parent 1	
Big Ears	1
Big Nose	1
Big Hands	1
Big Feet	1
Long Legs	1

Parent 2	
Small Ears	0
Small Nose	0
Small Hands	0
Small Feet	0
Short Legs	0

Child 1	
Small Ears	0
Small Nose	0
Small Hands	0
Big Feet	1
Long Legs	1

Child 2	
Big Ears	1
Big Nose	1
Big Hands	1
Small Feet	0
Short Legs	0

Figure 2.5: One-point Crossover



Small Ears	0
Big Nose	1
Small Hands	0
Big Feet	1
Long Legs	1

Figure 2.6: Mutation of child 1 from Figure 2.5

2.3.2.4 CONTROL PARAMETERS

When setting up a GA problem, the values of parameters such as the size of the population, the crossover rate, the mutation rate, the tournament size and the elitist size are modified. They should be treated with caution and are probably problem dependent [2.1] and according to Toropov [2.40] setting up is a "black art". DeJong [2.41] suggests a set of values for population size of 50-100, crossover rate 0.60 and mutation rate 0.001. Grefenstette and Fitzpatrick [2.42] recommended values of a population size 30, crossover rate 0.950 and mutation rate 0.01. Goldberg [2.9] suggest the optimal population size, $p_s = 1.65 \times 2^{0.21} \times$ *string length*. Ghasemi [2.29] reports that this might lead to excessively large population sizes with associated high computational costs. In practice, the parameters are chosen in a heuristic manner.

2.3.2.5 CONVERGENCE CRITERIA

Three possible convergence criteria are described here [2.1]. If one criterion is satisfied then the optimization process terminates. These criteria are:

1) When the percentage difference between the average value of all the designs and the best parent in a population (assuming non-penalized values) reaches a very small, specified value.
2) If the fittest design has not changed for 20 successive generations, or the difference of the fittest design of the current generation, and that of 20 generations before, is a small amount (typically 0.1)
3) If the total allocated number of generations is reached.

A problem discussed in detail by Goodman *et al.* [2.43] is of premature convergence stating that, "once a sub-optimal individual dominates the population, selection is likely to keep it there and prevent further adaptation within any practical timeframe." Goodman *et al.* [2.43] discuss the concept of *genetic drift*, which "is the modification of a population resulting from random events" and they go on to say that "under no selection pressure, a population will be dominated by genetic drift and converge. So whether selection or drift dominates, convergence is inherent in standard GAs and is the reason that standard GAs can't maintain different high fitness individuals in one population."

To prevent premature convergence Goodman *et al.* [2.43] suggest that there are two approaches. The first is to reduce the convergence speed by adjusting the selection phase and so allow the GA to do a wider search. The second approach is to use a parallel GA and allow parallel populations to evolve independently and share information between populations using a variety of techniques. Parallel GAs are discussed in detail by Goodman *et al.* [2.43] and include methods such as the injection island strategy [2.44].

2.3.2.6 RE-BIRTHING

In some optimization problems using continuous variables the resolution *res* may be insufficient to allow further improvement in the objective function over a set of iterations; re-birthing can be used to overcome this. The optimization problem may be started with a wide range for each design variable and a small bit-string length. Such a problem may be run with a GA until convergence is achieved. It is then possible to reduce the range for each variable and start the GA process again with the same bit-string length to provide a finer resolution.

2.3.3 HYBRID METHOD – MARS

The multipoint approximation method based on the response surface fitting (MARS) has been developed and used in a number of studies including [2.45, 2.46, 2.47, 2.48, 2.49]. The performance of the MARS method compared to a GA and the gradient-based optimizer DOT, is described in Chapter 5. A full description of the method can be found in Ref. [2.50], an overview, adapted from [2.50], is given below.

The MARS technique works by splitting the original problem into a sequence of simplified sub-problems; the optimum solution of each sub problem is the starting point for the next sub-problem, the bounds are changed and the process is repeated until convergence of the global optimization problem. The bounds or *move limits* for each approximation sub-region are determined by a “move limit strategy”. Each sub-problem is a ‘noise-free’, mid-range approximation of the original function. The points through which the approximations are fitted are determined by DoE methods (Section 3.4). In the current implementation, the approximations are either the linear or multiplicative approximations defined in Equations 3.3.2a and 3.3.2b respectively. In Ref. [2.51] Alvarez extends the method by using genetic programming (Section 3.3.2.1.5) to generate the approximations. Overall, the main factors, which influence the rate of convergence, are the choice of the structure of the approximate expressions, the choice of the DoE, the convergence criteria and the move limit strategy. For completeness, the theory of MARS is given below it is taken from Ref. [2.50, 2.52].

A general optimization problem can be formulated as

$$\text{minimize:} \quad F_0(\underline{x}) \tag{2.10a}$$

$$\text{subject to:} \quad F_j(\underline{x}) \leq 0; \qquad j = 1,\ldots,m \tag{2.10b}$$

$$A_i \leq x_i \leq B_i; \qquad i = 1,\ldots,n \tag{2.10c}$$

where m is the number of constraints, n is the number of design variables, $\underline{x}$ refers to the vector of design variables, $F_0(\underline{x})$ is the objective function to be minimized and $F_j(\underline{x})$ $(j > 0)$ are the constraints. A_i and B_i are the lower and upper bounds on the design variable x_i.

In order to reduce the number of calls for the response function evaluations and to lessen the influence of noise, the MARS replaces the optimization problem by a sequence of approximate optimization problems:

$$\text{minimize:} \quad \tilde{F}_0^k(\underline{x}) \tag{2.11a}$$

$$\text{subject to:} \quad \tilde{F}_j^k(\underline{x}) \leq 0 \qquad j = 1,\ldots,m \tag{2.11b}$$

$$A_i^k \leq x_i \leq B_i^k \qquad i = 1,\ldots,n \tag{2.11c}$$

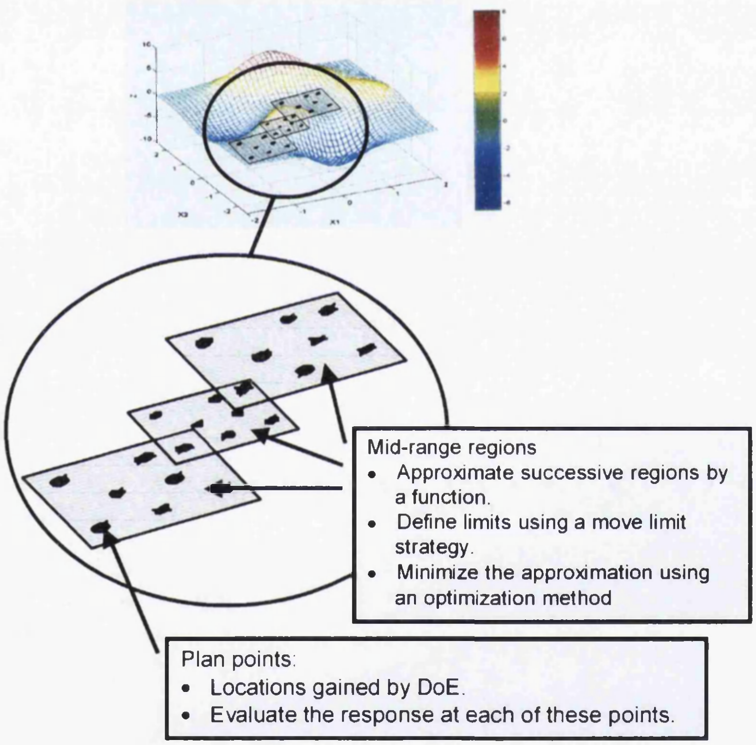
$$A_i^k \geq A_i \qquad i = 1,\ldots,n \tag{2.11d}$$

$$B_i^k \leq B_i \qquad i = 1,\ldots,n \tag{2.11e}$$

where k is the iteration number. The selection of the noise-free approximate response functions $\tilde{F}_j^k(\underline{x})$ $(j = 0,\ldots,m)$ is such that their evaluation is inexpensive as compared to the evaluation of the response functions F_j.

The approximate response functions are intended to be adequate in a current search sub-domain. This is achieved by appropriate planning of numerical experiments and use of move limits A_i^k and B_i^k. Since the functions in Equations 2.11a-e are chosen to be simple, computationally inexpensive, any conventional optimization method can be used to solve the problem. The obtained point $\underline{x}_*^k$ is then chosen as the starting point for the ($k+1$)-th step and the optimization problem to be solved i.e., Equation 2.11, is re-formulated with the new functions $\tilde{F}_j^{k+1}(\underline{x})$ $(j = 0,...,m)$ and move limits A_i^{k+1} and B_i^{k+1} $(i = 0,...,n)$. This is repeated until the convergence criteria are met.

Figure 2.7 illustrates the mid-range approximation, identifying the mid-range regions and plan points. At each of the plan points, the response is evaluated and the region is approximated using the responses. Further information of design space approximations is given in Section 3.3.



Mid-range regions
- Approximate successive regions by a function.
- Define limits using a move limit strategy.
- Minimize the approximation using an optimization method

Plan points:
- Locations gained by DoE.
- Evaluate the response at each of these points.

Figure 2.7: Conceptual representation of MARS

2.4 CONCLUDING REMARKS

This chapter has given an overview of optimization methods, with particular attention to the methods used elsewhere in the thesis. Much of the information contained is here for completeness and is considered as a 'black box'

2.5 REFERENCES

[2.1] Hinton E., Sienz J., Özakça M., 'Analysis and Optimization of Prismatic and Axisymmetric Shell Structures: Theory, Practice and Software', Springer Verlag, ISBN: 1852334215, 2003.

[2.2] Dornberger, R., Büche., Stoll, P. “Multidisciplinary optmization in turbomachinery design.” ECCOMAS, 2000.

[2.3] Vanderplaats G.N., “DoT User Manual”, VMA, Engineering, USA, 2000.

[2.4] Powell M.J.D., “VMCWD: A Fortran subroutine for constrained optimization”, DAMTP 1982/NA4, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge, UK, 1982.

[2.5] Chamberlain R.M., Lemarechal C., Pedersen H.C. and Powell, M.J.D., “The watchdog technique for forcing convergence in algorithms for constrained optimization”, Mathematical Programming Study, 16, 1-17, 1982.

[2.6] Schittkowski K., “Non-linear Programming Software”, software manual, Department of Mathematics,Universitat Bayreuth, Germany, 1991.

[2.7] Svanberg K., “The method of moving asymptotes – a new method for structural optimization”, International Journal Numerical Methods in Engineering, 24, pp. 359-373, 1987.

[2.8] Lasdon L.S., Warren A.D., Jain A. Ratner M., Design and testing of a generalized reduced gradient code for non-linear programming”, ACM Trans. Math. Softw., 4(1), 34-50, 1978.

[2.9] Goldberg D.E. “Genetic algorithms in Search, Optimization, and Machine Learning”,Reading,Mass.:Addison-Wesley,1989.

[2.10] Fogel, D.B., “Applying evolutionary programming to selected travelling salesman problems”, Cybernetics and Systems, 24(1), 27-36, 1993.

[2.11] Bäck T., Hoffmeister F. and Schwefel H.-P., “A survey of evolution strategies”, in Belew, R.K. and Booker, L.B. (eds), Proceedings of the 4th International Conference on Genetic Algorithms (ICGA IV), 2-9, Morgan Kaufman Publishers Inc., San Diego, 1991.

[2.12] Kennedy, J. and Eberhart, R.C., “Particle swarm optimization”, Proceedings of IEEE International Conference on Neural Networks, IV, 1942-1948, Piscataway, NJ, 1995.

[2.13] Kirkpatrick S., Gelat C.D., Vecchi M.P. “Optimization by Simulated Annealing.”Science, Vol. 220, pp.671-680, 1983.

[2.14] Belegundu, A.D., Chandrupatla, T.R. “Optimization Concepts and Applications in Engineering”, Prentice-Hall 1999.

[2.15] Edgar, T.F., Himmelblau, D.M. “Optimization of Chemical processes”, McGraw-Hill 1989.

[2.16] Bletzinger K.U., “Formoptimierung von Flächentragwerken” Ph.D. thesis, Bericht Nr.11, Institut für Baustatik der Universität Stuttgart, Stuttgart, Germany, 1990.

[2.17] Sienz J., “Integrated structural modelling, adaptive analysis and shape optimization”, PhD Thesis, University of Wales Swansea, 1994.

[2.18] Vanderplaats G.N., Moses F., “Structural optimization by methods of feasible directions”, Computers and Structures, 3, 739-755, 1973.

[2.19] Fleury C., “CONLIN: An efficient dual optimizer based on convex approximation concepts”, Structural Optimization, 1, pp. 81-89, 1989.

[2.20] Stoer J., “Foundations of recursive quadratic programming methods for solving non-linear programs, in Schittkowski, K., (ed), Computational Mathematical Programming, NATO ASI Series, Series F:Computer and Systems Sciences, 15, Springer, 1985.

[2.21] Lee S.J., “Analysis and Optimization of Shells”, PhD thesis, C/Ph/223/98, ADOPT Group, Department of Civil Engineering, University of Wales Swansea, Swansea, UK, 1988.

[2.22] Hadley G., “Nonlinear and Dynamic Programming”, Addison-Wesley, Reading, USA, 1964.

[2.23] Fox R.L., “Optimization Methods for Engineering Design”, Addison-Wesley, Reading, USA, 1971.

[2.24] Schittkowski K., Zillober C., Zotemantel R., “Numerical comparison of non-linear programming algorithms for structural optimization”, Structural Optimization, 7,1-19, 1994.

[2.25] Vanderplaats G.N., “Numerical Optimization Techniques for Engineering Design”, McGraw-Hill, 1984.

[2.26] Parsopoulos K.E., Vrahatis M.N., “Recent approaches to global optimization problems through particle swarm optimization” natural computing, 1: 235–306, 2002.

[2.27] Holland J., “Adaptation in Natural and Artificial Systems”, University of Michigan Press, 1975.

[2.28] Rajeev, S. and Krishnamoorthy, C.S., Discrete optimization of structures using GA, J. S. Engng., ASCE, 118 (5), 1233-1250, 1992.

[2.29] Ghasemi, M.R. Structural Optimization of Trusses and Axisymmetric Shells using Gradient based Methods and Genetic Algorithms, PhD thesis, C/Ph/207/96, ADOPT Research Group, Civil Engineering, University of Wales Swansea, 1996.

[2.30] Ghasemi, M.R. and Hinton, E., Truss optimization using genetic algorithms, in Topping, B.H.V. (eds) Proceedings on Advances in Computational Structures Technology, Civil Comp Press, Edinburgh, 59-75, 1996.

[2.31] Gurdal, Z., Haftka, R., Hajela, P. "Design and optimization of laminated composite materials" New York ; Chichester : Wiley, 1999.

[2.32] Bäck T., Rudolph G., Schwefel H.-P. "Evolutionary Programming and Evolution Strategies: Similarities and Differences", Fogel D.B., Atmar W., ed: Proc. 2nd Conf. Evolutionary Programming, pp.11-22, EP Soc., San Diego, 1993.

[2.33] Fogel L.J., Owens A.J., Walsh M.J., "Artificial intelligence through simulated evolution", Wiley, Chichester, UK, 1996.

[2.34] Schutte J.F., Fregly B.J., Haftka R.T., George A., "A parallel particle swarm optimizer", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.245-246, 2003.

[2.35] Venter G., Haftka R.T., Sobieski J. S., "Robust design using particle swarm and genetic algorithm optimization", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.301-302, 2003.

[2.36] Wilke D.N., Groenwold A.A., "On constrained optimization of structural systems using particle swarms", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.91-92, 2003.

[2.37] Yu H., Zhaoying Z., " The particle swarm optimization algorithm and its application in multidisciplinary optimization", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.315-316, 2003.

[2.38] Langley D.S., "Genetic algorithm focussed comparative optimization studies for a broad scope of engineering applications", PhD Thesis, University of Wales Swansea, 2003.

[2.39] Merriam-Webster online dictionary. http://www.m-w.com

[2.40] Personal communication from Prof. Toropov V.V. E-mail: toropov@altair.com, 2003.

[2.41] DeJong K. "An Analysis of the Behaviour of a Class of Genetic Adaptive Systems", PhD thesis, Dept. of Computer and Communications Sciences, Univ. of Michigan, 1975.

[2.42] Grefenstette J.J. Fitzpatrick J.M., Genetic search with approximate function evaluations", in Proceedings of First International Conference on Genetic Algorithms and Their Applications, L. Erlbaum Inc., New Jersey, 112-120, 1985.

[2.43] Goodman, E.D., Punch, W.F. & Lin, S. "Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach", Sixth IEEE Parallel and Distributed Processing, pp 28-37, 1994.

[2.44] Parmee I., Vekeria H., “Co-Operative Evolutionary Strategies for Single Component Design,” Proc. 7th Int. Conf. Genetic Alg., T. Baeck, ed., Morgan Kaufmann, San Francisco, 529-536, 1997.

[2.45] Toropov V.V., "Simulation approach to structural optimization". Structural Optimization, 1, No 1, pp. 37-46, 1989.

[2.46] Toropov V.V., Filatov A.A., Polynkin A.A., “Multiparameter structural optimization using FEM and multipoint explicit approximations”, Structural Optimization, 6, 7-14,1993

[2.47] Toropov V.V., van Keulen F., Markine V.L., de Boer H., “Refinements in the multi-point approximation method to reduce the effects of noisy responses”, 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue WA, September 4-6 1996, Part 2, pp. 941-951,1996.

[2.48] Van Keulen F., Toropov V.V., “New developments in structural optimization using adaptive mesh refinement and multi-point approximations”, Engineering Optimization, v. 29, 217-234, 1997.

[2.49] Sienz J., Bates S.J., Langley D.S., Pittman J.F.T. “Comparison of globally converging optimization with mid-range approximation based optimization applied to slit die design” Proc. 3rd ASMO UK / ISSMO conf., Harrogate, UK, 2001.

[2.50] Toropov, V.V.; Alvarez, L.F. Approximation model building for design optimization using genetic programming methodology. Paper AIAA-98-4769, 7th AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization, St. Louis (USA), September 2-4, 1998, part 1, pp. 490-498, AIAA, 1998. ISBN 1-56347-273-2, 1998. http://www.brad.ac.uk/staff/vtoropov/burgeon/b_mars.htm, 1998.

[2.51] Alvarez L.F., “Design optimization based on genetic programming: Approximation model building for design optimization using the response surface methodology and genetic programming”, PhD Thesis, Bradford University, UK, 2000.

[2.52] Markine V.L., “Optimization of the dynamic behaviour of mechanical systems”, PhD Thesis, Delft University of technology, 1999.

CHAPTER 3

OPTIMIZATION MODELLING

Summary: this chapter describes the methods used to parameterize a design problem for use in an optimization process. The parameterization defines the design variables, constraints and boundary conditions. Methods for approximating and analysing the design space using the parameters are also described in this chapter. The topics covered include material distribution methods – topology, shape and size optimization, approximation methods, design of experiments and multi-objective optimization. Two newly developed techniques are presented; the first is a method for formulating an optimal Latin Hypercube design of experiments using a permutation genetic algorithm; the other is a multi-objective method for determining a uniform Pareto set of solutions. The method is based on the physical programming method, and is called the Pareto front marching method.

3.1 INTRODUCTION

Section 3.1: Gives an introduction to the chapter, explaining what optimization modelling is and where and how the various methods fit into the optimization process.

Section 3.2: Describes the material distribution methods: shape, size and topology optimization, and how they are combined in a design process.

Section 3.3: Looks at the methods for design-space approximating. These methods are used in optimization for two main reasons: (i) to minimize the number of response evaluations, and (ii) to reduce the effect of noise.

Section 3.4: Describes the design of experiments (DoE) methods used elsewhere in the thesis. Particular attention is paid to a new technique developed in this work for the formulation of the optimal Latin Hypercube DoE.

Section 3.5: Looks at multi-objective optimization and the concept of Pareto optimality to represent the trade-off between competing objective functions. Particular attention is paid to solving robust design problems where there are two competing objectives. Several methods are investigated, including the Normal boundary intersection method; the Physical programming method, and a new method is developed, called the Pareto front marching method.

Section 3.6: Gives some concluding remarks.

Section 3.7: Lists the references used in the chapter.

This chapter is concerned with optimization modelling, the second column of the three-column concept in Figure 1.3. Optimization modelling is used to parameterize a problem, by defining the design variables and the constraints; it also includes methods for approximating the design space using those parameters. In developing the optimization model one or a combination of the methods in the following broad categories can be used:

- Material distribution methods- topology, shape and size optimization
- Design space approximations.
- Design of experiments.
- Multi-objective optimization.

This chapter describes each of the above categories. It is set out as follows: firstly, the general theory of the topic is discussed, then secondly, more detailed information is given on the specific methods used in the thesis. Thirdly, where it has been necessary to develop the methods further, a full description is given.

3.2 MATERIAL DISTRIBUTION METHODS

Material distribution methods [3.1] are methods for finding the optimum layout of a structure, including information on the topology, shape and size of the structure. The methods are used to parameterize a problem such, that changes in the design variables will alter the material distribution of the structure according to the method used. There are three main material distribution methods and the effect of altering the design variables in each is as follows:

- **Topology optimization**: allows adding/removal and re-distribution of material in a structure.
- **Shape optimization:** changes the shape of a structure.
- **Size optimization:** changes thickness and area distributions in a structure.

"The sizing and shape optimization methods may lead to sub-optimal results as they suffer from the problem of not necessarily having an optimum starting topology" [3.2]. In structural design, a commonly used three-stage design optimization process, called FIDO in Ref. [3.3], is as follows:

1) Topology optimization: to get the optimal locations, sizes and shapes of holes and the connectivity of the structure.
2) Structural redefinition: to extract the geometry from the topology optimization.
3) Size and shape optimization to *fine-tune* the topology.

An example application of this process, including consideration of manufacturing restrictions, is given in Chapter 6, for the design of an automotive component.

3.2.1 TOPOLOGY OPTIMIZATION

Topology optimization is a form of structural optimization that makes it possible to produce an optimal material layout of a structure for a given set of loads and constraints. The parameterization of the design space allows the removal/adding and the re-distribution of material in a structure. The performance is based on a response, e.g. the stiffness. The aim of the optimization is to modify the design variables, i.e. the material distribution, to find the solution that is best able to cope with the boundary conditions. The objective is to have no restriction on the final form of the structure so that, for example, as the structure changes holes may appear. It is only the loads and constraints that are applied to the structure that are necessary for the method; the actual form of the structure is not. However, a design space in which the structure can evolve must be defined. Using topology optimization provides possible answers to problems like: "A cantilever is constrained and loaded as shown in Figure 3.1(a), what is the optimum structure to support the load when the structure is constrained as shown?" The problem in Figure 3.1(a) has a unique theoretical solution due to Michell [3.4], it consists of a tie and a strut making a 90° connection at the load, and a 45° internal angle to the vertical as can be seen in Figure 3.1(b).

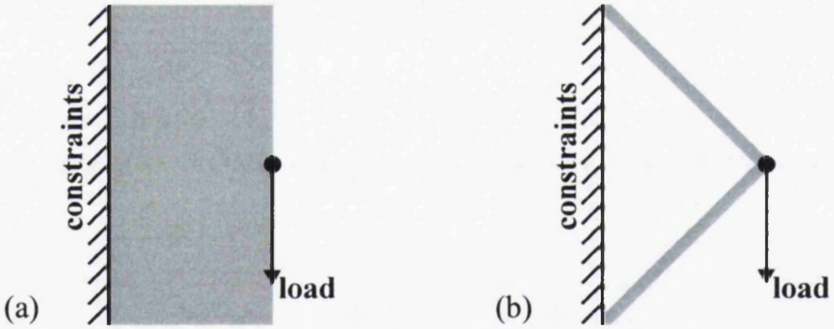


Figure 3.1 (a) Cantilever showing constraints and loads; (b) the optimal structure

Following the notation in [3.5] there are two main branches of topology optimization: *material removal* and *material re-distribution* methods. Material removal methods are typically based on evolutionary methods; material re-distribution methods are more mathematically rigorous and have their origin in the homogenization method. A description of some of the methods is presented in Sections 3.2.1.1 and 3.2.1.2, and a discussion on numerical instabilities is given in Section 3.2.1.3. Rozvany [3.6, 3.7], Bendsøe and Sigmund [3.1] give extensive reviews and descriptions of the topology optimization methods.

3.2.1.1 MATERIAL REMOVAL METHODS

Material removal methods are typically based on evolutionary or 'hard-kill/soft-kill' methods and include adaptive biological growth (ABG) Mattheck *et al.* [3.8], and evolutionary structural optimization (ESO) Xie and Steven [3.9]. These methods generate structural topologies by iteratively 'removing' elements that are under-used according to some response parameter, such as stress, or compliance; or by adding elements around existing elements that are over-used. The starting point is usually a continuum discretized into an FE mesh, which is then analyzed to determine the response parameter for each element. The removal or addition of elements can be done in two ways: (i) by deletion or creation of new elements and hence creating a new stiffness matrix or (ii) by alteration of the elastic modulus of an element according to a linear relationship (soft-kill), a step function (hard-kill) or by some other function; thereby not deleting elements or creating new ones but by merely switching them 'off' or 'on'. A summary of some of the methods in this area is given below:

3.2.1.1.1 ADAPTIVE BIOLOGICAL GROWTH (ABG)

The idea of this method is to simulate biological growth by volumetrically expanding or contracting a structure according to its stress distribution. This is achieved by assigning varying temperatures throughout the FE model according to the stress distribution; areas of low stress are assigned temperatures below the ambient temperature, causing contraction. Similarly, highly stressed areas are assigned higher temperatures causing expansion. At each iteration, the topology of the FE model is updated and two FEA runs are required – one for a static and one for thermal stress analysis. Querin and Lencus [3.10] extend ABG to cope with multiple load cases and they refer to the method as the Biological Growth Structural Optimisation method (BGSO).

3.2.1.1.2 ORIGINAL ESO

In the original ESO method, elements can only be removed and not inserted. The criterion for removing an element is as follows

$$F_e \leq RR_i \times F_{\max}, \tag{3.2.1}$$

where F_e is the response parameter of an element e, $F_{\max}$ is the maximum value of the response parameter in the structure and RR_i is the rejection ratio. Once the elements have been removed, an FE analysis of the structure is done, and the iterative removal of elements is repeated until a steady-state is reached, i.e. no more elements are removed. At this point an increased rejection ratio is used. This is given by

$$RR_{i+1} = RR_i + ER, \tag{3.2.2}$$

where ER is the evolution rate. Typically, initial values for ER and RR_0 are 1%.

The original ESO has the problem, that if the response distribution, e.g. the stress distribution, changes considerably then, in order to achieve an optimum solution, elements need to be added.

It should be noted, that the response parameter used in original ESO and all of its derivatives, is not only restricted to using a stress or compliance criterion. Other criteria have been used, such as heat conduction [3.11], thermal stress [3.11] and natural frequency [3.12].

3.2.1.1.3 Bi-directional ESO (BESO)

BESO was developed by Querin *et al.* [3.13]. It addresses the problem with original ESO by allowing the addition *and* removal of elements; it is combination of the additive ESO method (AESO) and the original ESO. In BESO elements are removed from regions if they satisfy Equation 3.2.3 or added to the structure if they satisfy Equation 3.2.4.

$$F_e \leq RR \times F_{\max} \tag{3.2.3}$$

$$F_e \geq IR \times F_{\max} \tag{3.2.4}$$

where RR and IR are the rejection and inclusion ratios respectively, for further details see Querin *et al.* [3.13].

3.2.1.1.3 Morphing ESO (MESO)

MESO was developed by Querin *et al.* [3.14]. It was introduced to allow for both the partial removal of material from a structure, and the modification of the material properties. The main difference between the original ESO and MESO is, that unlike with the original ESO, the elements can be either on or off, i.e. the design variable set is discretized using a {0,1} binary representation, MESO uses an increased discrete variable set. This means, that instead of removing or adding an element totally in one step, the element can be removed gradually by incrementally altering its properties to be of less strength, thickness or density. In this sense this method could also be regarded as a material re-distribution method. The process is terminated when a minimum performance index value is reached, or when a desired volume or mass is reached.

3.2.1.1.4 GROUP ESO (GESO)

GESO was developed by Lencus *et al.* [3.15]. The idea is to remove groups of elements instead of single elements; otherwise it works in the same manner as the original ESO. It is used to find the optimal configuration of components within a structure.

3.2.1.1.5 NIBBLING ESO (NESO)

NESO is based on the original ESO concept, but only removes elements that are on the boundaries. Since, only the structural boundary is optimized, the method can be considered as a shape optimization method.

3.2.1.1.6 INTELLIGENT CAVITY CREATION (ICC)

ICC was developed by Kim *et al.* [3.16]. It is a method for controlling the number of cavities produced with the original ESO. In this method, NESO is used to modify the structural boundary and when the need for a cavity arises, the original ESO is used. The point at which the original ESO is applied is determined by the response distribution. At each iteration, the ratio of the minimum response parameter in the whole structure to the minimum response parameter on the boundary is calculated. This ratio is called the Insert Cavity Ratio (ICR):

$$ICR = \frac{F_{\min}}{F_{boundary,\min}} \tag{3.2.5}$$

where $F_{\min}$ and $F_{boundary,\min}$ are the minimum values of the response parameter in the whole structure and on the boundary respectively. If ICR is less than some specified value, a cavity is created using the original ESO, thereby creating another boundary.

NESO is then applied until there is a need for another cavity or the optimization termination criteria are reached.

3.2.1.1.7 ESO USING THE BOUNDARY ELEMENT METHOD AND B-SPLINES

Cervera and Trevelyan [3.17] introduced an alternative approach to ESO. It was introduced to cope with some shortcomings of ESO, namely jagged edges and checkerboard patterns resulting from mesh dependency (Section 3.2.1.3). The method uses the boundary element method (BEM) instead of the traditional FEM and uses B-spline curves to represent the boundary shape. The approach uses the control points of the B-spline as the design variables and the boundary smoothness is achieved by moving the control points instead of deleting and adding elements. The method allows for insertion and merging of holes.

3.2.1.1.8 GENETIC PROGRAMMING AND GENETIC ALGORITHMS

Some other recent studies of interest are

- Zheng *et al.* [3.18] who use genetic programming (Section 3.3.2.1.5) to carry out topology optimization of discrete structures;

- Woon *et al.* [3.19] use a 'Multi-GA system' for the topology optimization whereby two GAs are run in parallel to each other; the first is used to optimize the external boundary whilst the other simultaneously optimizes the internal topology.

3.2.1.2 MATERIAL RE-DISTRIBUTION METHODS

Material re-distribution methods are more mathematically rigorous than the material removal methods; they are based on the homogenization method developed by Bendsøe and Kikuchi [3.20]. "Initial work on numerical methods for topology design of continuum structures was based on using composite materials as the basis for describing the varying material properties in space. This approach has been named the *homogenization method.*" [3.1]

The formulation of the topology optimization problem is as follows:

$$\begin{aligned} &\text{minimize: } F(\rho) \\ &\text{Subject to: } V = \sum_{i=1}^{n} \rho_i v_i \leq V^t \\ &\qquad 0 \leq \rho_i \leq 1, \qquad i = 1,\ldots,n \end{aligned} \tag{3.2.6}$$

where $F(\rho)$ is the objective function, ρ_i is the relative elemental density, v_i is the elemental volume, V^t is the target volume, n is the total number of elements.

As an aside, Chiandussi *et al.* [3.21] carry out topology optimization where the volume of the optimal solution depends on the imposed static (displacement, stress, stiffness) and dynamic (natural frequency) constraints and does not need to be specified *a priori.*

Using the homogenization method in a topology optimization loop requires that the material parameters of each FE are design variables. So, if there are one million designable elements in the package space then there are at least one million design variables. Typically, the topology optimization problems using homogenization are solved using optimality criteria methods or a dual-type method such as CONLIN by Fleury [3.22] or MMA by Svanberg [3.23].

In the homogenization method, the structure is represented by a porous continuum with microstructures. Each element is made up of an infinite number of microstructures. The three most common types of microstructures used are rectangular microscale voids, ranked layered material cells and a solid isotropic microstructure with penalization for intermediate densities (SIMP).

In order to apply the homogenization method in an optimization loop it is necessary to parameterize the microstructure with a set of design variables. Modification of the design variables will alter the microstructure, hence an updated homogenized elasticity matrix, D^H, is required. Depending on the parameterization used, the updated D^H can be determined analytically or numerically, either way it is essential that its determination is fast and efficient; especially when one considers that each element in the package space, with its own set of design variables, can have a different microstructure (but it is constant throughout each element) and a different D^H. Hence, the formulation of D^H is determined *a priori* for the microstructure being used. In the case of the rectangular microscale voids and SIMP microstructures it is necessary to determine approximations of D^H with respect to the design variables employing a set of FE calculations; with the ranked layered material cells an exact analytical expression can be derived.

It should be noted, that in choosing the microstructure, one of the important features is that changes in the design variables permit the cell to be either void or solid, i.e. the density of the material can have a range from zero to one.

3.2.1.2.1 Microstructure with a rectangular void

Figure 3.2.1 shows a typical rectangular void microstructure that is used. It can be seen that the continuum is divided into FEs (which in this case are triangular) and each FE is made up of an 'infinite' number of unit cells containing rectangular voids. The design variables for each cell are the width a, and the height b. The orientation of the microstructure θ, is achieved by aligning it with the principal stress. The volume of solid in the element is

$$1 - ab \tag{3.2.7}$$

and so the density is a function of a and b

$$\rho = (1 - ab)\rho_s \tag{3.2.8}$$

where ρ_s is the density of the solid. It can be seen, that if a and b are zero then the cell is solid or if a and b are one then the cell is void. In the three-dimensional case the voids are hexahedral and require an additional design variable c, to define its depth.

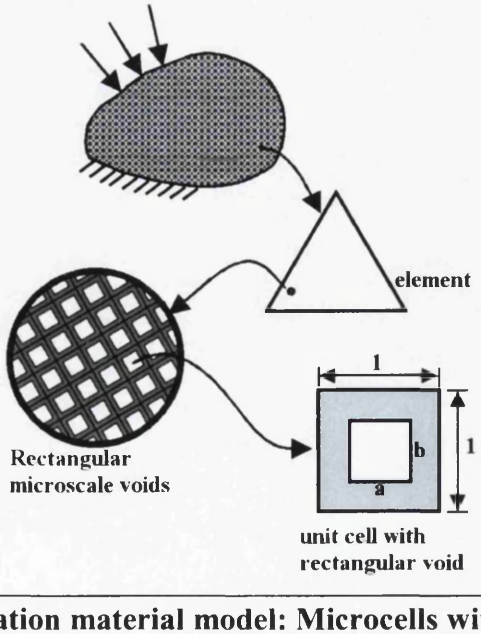


Figure 3.2.1 Homogenization material model: Microcells with rectangular voids

3.2.1.2.2 RANKED LAYERED MICROSTRUCTURE

Figure 3.2.2 shows a rank-1 and rank-2 layered material cell. Again, it can be seen that the continuum is divided into FEs (which in this case are triangular) and each FE is made up of an 'infinite' number of unit cells containing ranked layered material. The rank-1 material layer consists of alternating layers of solid material and voids (which are modelled using a very soft, flexible material). The relative densities of the solid and soft layers are denoted by γ and $1-\gamma$ respectively. The rank-1 material can be used to construct higher ranked materials, such as the rank-2 material which consists of repeated layers of solid and rank-1 material with relative densities of μ and $1-\mu$ respectively. The layers of different ranks are orthogonal to each other. The design variables for each cell are the width γ, and the height μ. Again the orientation of the microstructure θ, is determined using principal stresses.

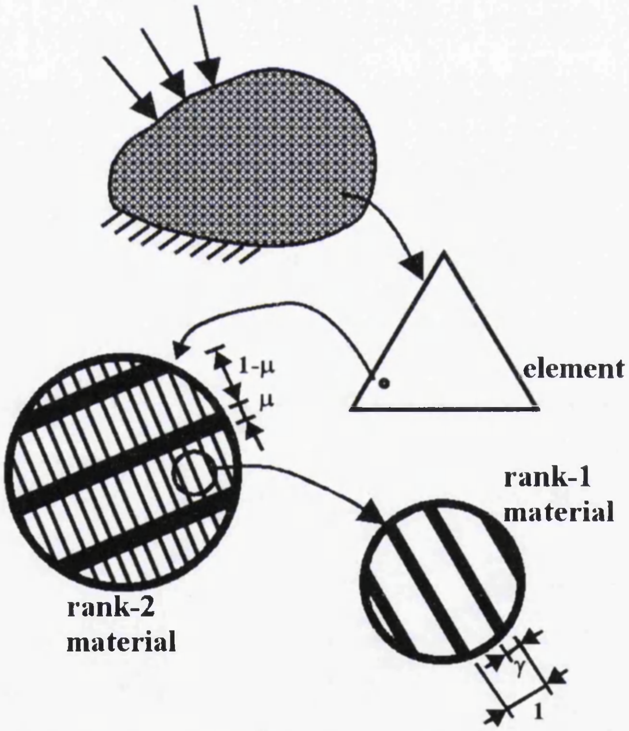


Figure 3.2.2 Homogenization material model: Ranked layered material cells

The density of the rank-2 composite is

$$\rho = (\gamma + \mu - \gamma\mu)\rho_s \tag{3.2.9}$$

meaning, that as with the rectangular void, it is possible to have a solid or a void microstructure.

3.2.1.2.3 SIMP

Another microstructure formulation is the *artificial density approach* proposed by Bendsøe [3.24], now widely known as the SIMP method (Solid Isotropic Microstructures with Penalization for intermediate densities). In this method the microstructures are solid and isotropic and there is only one design variable, the density. This again, ranges from zero to one (void to solid) where intermediate densities represent fictitious material.

In Chapter 6, the SIMP method is used to carry out topology optimization of an automotive component. "SIMP is becoming generally accepted in topology optimization as a technique of considerable advantages"[3.6].

3.2.1.2.4 PROBLEM FORMULATION

Ideally, from a practical engineering point of view, the material distribution problem would be formulated as a discrete problem, i.e. the microstructure is either void or solid. This has the problem that "the optimization results will be strongly dependent on the mesh discretization and the integer formulation comprises many artificial local minima."[3.2]. In order to overcome this, the problem is 'relaxed' by re-formulating it as a continuous one, thereby allowing for porous areas. For example,

$$\rho = \xi \rho_s \tag{3.2.10}$$

where ξ is a continuous parameter and $0 \le \xi \le 1$ and ρ_s is the density of the solid. The elasticity matrix is then calculated by

$$D = \xi D_s . \tag{3.2.11}$$

where D_s is the elasticity matrix of the solid. In order to enforce the solid/void ideal and limit the number of porous regions, the intermediate values of density are penalized so that

$$D = \xi^p D_s , \tag{3.2.12}$$

where p is a penalization factor. It is greater than one, it usually takes a value between one and three. So, for the material model consisting of a microstructure with rectangular voids

$$D = (1 - ab)^p D^H , \tag{3.2.13}$$

and for the SIMP material model

$$D = \rho^p D^H . \tag{3.2.14}$$

Further details on this subject can be found in Hassani and Hinton [3.2] or Bendsøe and Sigmund [3.1].

3.2.1.3 NUMERICAL INSTABILITIES

Sigmund and Petersson [3.25] and Bendsøe and Sigmund [3.1] give a review of the procedures to deal with the numerical instabilities in topology optimization. They separate the instabilities into three categories: checkerboarding, mesh dependencies and local minima. They also identify a number of methods to overcome them. These methods are briefly discussed below for completeness together with a discussion of the minimum member size control, which is used in Chapter 6.

3.2.1.3.1 PROBLEMS

Checkerboarding – refers to the formation of regions of alternating solid or void elements ordered in a checkerboard-like fashion (see Figure 3.2.3). This causes highly discontinuous density variations, leading to problems when numerically modelling the stiffness of the checkerboards and during the solution process of the systems of equations. [3.25]

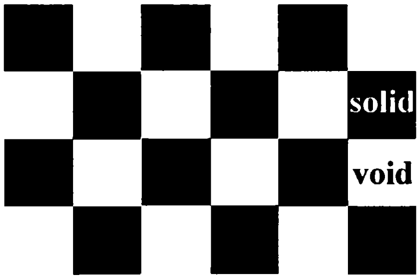


Figure 3.2.3: Elemental checkerboarding

Local minima – refers to the problem of obtaining different solutions to the same discretized problem when choosing different algorithmic parameters such as the starting point. [3.25]

Mesh dependencies – this refers to the problem of obtaining qualitatively different solutions for different mesh-sizes or discretizations. [3.25]

3.2.1.3.2 SOLUTIONS

Local minima problems arise due to many problems having multiple optima. Bendsøe and Sigmund [3.1] state that continuation methods 'must be applied to ensure some sort of stable convergence towards reliably good designs'. The idea of continuation methods is to gradually change the optimization problem from an (artificially) convex problem to the original non-convex problem. For example, for SIMP this can be done by iteratively increasing p (the penalization factor), starting from $p = 1$ until the final design is reached.

Suggested techniques to overcome the checkerboarding problems are:

- Smoothing the optimal solution (including the checkerboards) by using image processing, this ignores the underlying problem.
- Using higher-order finite elements, this substantially increases the CPU time.
- Using patches by introducing 'super elements' to damp the appearance of checkerboards, this does not entirely remove them.
- Using a filter by making the design sensitivity of an element depend on a weighted average over the element itself and a 'fixed neighbourhood'.

"In general, techniques used to achieve mesh independent solutions overcome the checkerboard problem"[3.26]. Suggested techniques to overcome the mesh dependence problems are to use a 'relaxation' of the problem as discussed above for the homogenization method or a 'restriction method' such as perimeter control, mesh independent filtering or density slope control.

Perimeter control (Ambrosio and Buttazzo [3.27], Haber *et al.* [3.28]) is a method to constrain the number of holes that can appear in a domain. It does this by putting a constraint on the perimeter length, which is (approximately) calculated by the sum of the lengths/areas of all the boundaries. The main problem with this method is that "it is not easy to define the proper bound on the perimeter that leads to a desired simplicity of topology that reflects a specific manufacturing need"[3.26].

Mesh independent filtering (Sigmund [3.29]) is a rule used for modifying the sensitivity of the objective function to ensure that wherever a member is formed, the radius is greater than a minimum value $r_{\min}$, which is half of the predetermined minimum member diameter; increasing the sensitivity of an element if the elements within a distance, $r_{\min}$, of the element have higher sensitivities does this. Zhou *et al.* [3.26] note, that an undesired effect of this method is that members in the final solution involve layers of elements with intermediate densities.

Petersson and Sigmund [3.30] introduced the density slope control method to address the issue of numerical problems caused by highly discontinuous density variations. The method introduces a constraint on the local gradient of the slope of element densities. It requires $2n$ and $3n$ additional linear constraints for 2D and 3D structures respectively, "making the approach computationally prohibitive for practical applications"[3.26].

"From an engineering point of view, the mesh dependency of the design may not be as significant as the concern about the manufacturability of the resulting topology. This means that the designer may not worry about the fact that different mesh densities may result in different final solutions. Instead, it may be more important to be able to control the size of the members in the final topology, and therefore, the degree of simplicity of the design. Besides manufacturing considerations, very thin members are unstable under compression" [3.26].

Based on the density slope control approach (discussed above) Zhou *et al.* [3.26] developed a minimum member size control (MMSC) and implemented "an efficient algorithm to treat the additional constraints very efficiently, by exploring their special characteristics" at "almost no extra computational cost." The constraint used in that work is as follows:

$$|\rho_i - \rho_k| \leq \frac{dist(i,k) \times (1.0 - \rho_{\min})}{r_{\min}}, \tag{3.2.14}$$

where $dist(i,k)$ is the centroid to centroid distance between adjacent elements i and k and $\rho_{\min}$ is the threshold density interpreted as a void in the final solution. This condition guarantees that whenever an element reaches the density of 1.0, the member connected to this element has a diameter of at least the value of the predetermined minimum member diameter.

The MMSC algorithm is implemented in the commercial optimization software, OptiStruct [3.31]. It is used to produce a smooth topology that can overcome the none-smoothness of the FE mesh because "you do not want to have ribs developing that are dominated by 'zigzagging' element surfaces. So, the wider the rib is, the less significant the none-smoothness of the boundary layer of elements is."[3.32] A problem with this method is, that even though it may reduce the number of cavities it does not prevent their creation. When considering manufacturing feasibility this is clearly undesirable. To address this issue, Zhou *et al.* [3.33] introduced manufacturing constraints which have the effect that only cavities that are open and aligned with the sliding direction of the die are permitted in the optimization.

A study into the effect, that MMSC has on the final topology generated using two different mesh densities, is carried out in Chapter 6 of this work. A comparison of the results (shown in Figures 6.22 and 6.23) shows, that when using the default minimum member size, i.e. three-times the average element size in the mesh, significantly different topologies are generated; when specifying the same absolute minimum member size for the two different mesh densities then roughly the same solution results.

3.2.2 SHAPE AND SIZE OPTIMIZATION

Shape optimization is a boundary variation method. It is used to parameterize a structure into a set of design variables, which when altered, change the outline shape of a structure, but the topology of the structure remains the same. For example, the boundary can be represented using splines (in 2D) or Coon's patches (in 3D) and the parameterization is done using the control points of the splines or surfaces. Movement of the control points will require modifications to the FE mesh that is used. These modifications can be achieved by either re-meshing or by translating the FE nodes at the control points and assigning perturbation vectors to nodes in the region of the control point movement.

Size optimization refers to the determination of specific geometric dimensions such as the thickness of a shell, rod member area, beam cross-section, spring, and mass properties.

Two industrial examples of using shape optimization have been carried out in this work investigating various aspects of shape optimzation. In Chapter 5, a cubic spline is used to represent the boundary of a structure and five design variables are used to alter its shape. In Chapter 6, after topology optimization is done, the FE model is parameterized with three shape variables.

3.3 DESIGN SPACE APPROXIMATION

The 'design space' is a hypothetical region where all of the possible designs for solving a problem exist. Each design has a response according to some performance characteristic. So if the response for every one of the infinite positions in the design space is known, then the resulting 'response' surface is formulated. E.g. a response surface for a design space consisting of two design variables, x_1 and x_2, can be seen in Figure 3.3.1(a), taken from [3.34], where the response is Z. The objective of the optimization is to find the combination of the design variables corresponding to the minimum (or maximum) value on the response surface. This can be done quite easily by inspection for the response surface in Fig 3.3.1(a). However, for more than two design variables this visualization is not possible. In an ideal world every possible solution could be computed and hence the response surface can be easily formulated. But often, the CPU time required to carry out one simulation of the system performance may take many hours or days. In an industrial environment this may be computationally prohibitive as thousands of simulations may be required to solve an optimization problem. If an accurate, noise-free approximation of the response surface is formulated, it is not always necessary to carry out as many of the computationally expensive simulations to solve the problem. Golovidov [3.35] states, that by using a "mathematical model to approximate the behaviour of the complex simulation tool and using it during optimization for gradient calculations and finding the next optimum search direction, can reduce the number of "exact" analyses by 2 to 10 times."

Overall, design space approximations, also known as *metamodels* (models of models), are often used in optimization for two main reasons: (i) to minimize the number of response evaluations, and (ii) to reduce the effect of noise. In this work, design space approximations are used in two problems:

- In Chapter 5, the optimization method, MARS (Chapter 2), is used in the optimization loop. This method replaces the original optimization problem by a succession of mid-range design space approximations.

- In Chapter 6, the expensive FE simulation model, is replaced by several second-order, globally approximating response surfaces, which are then used in an optimization loop to carry out shape optimization.

Design space approximation can be split into three bands according to Barthelemy and Haftka [3.36]: Global approximations, mid-range approximations, and local approximations. Local approximations are valid near the design point in the design space. Global approximations are valid for the entire design space. Mid-range lies between these two extremes. Figure 3.3.1(a) illustrates a response surface with two variables, x_1 and x_2, and it shows the differences between local (Figure 3.3.1(b)), mid–range (Figure 3.3.1(c)) and global (Figure 3.3.1(d)) approximations. The grey areas in Figures 3.3.1(b) to 3.3.1(d) are the regions in which the response surface is approximated.

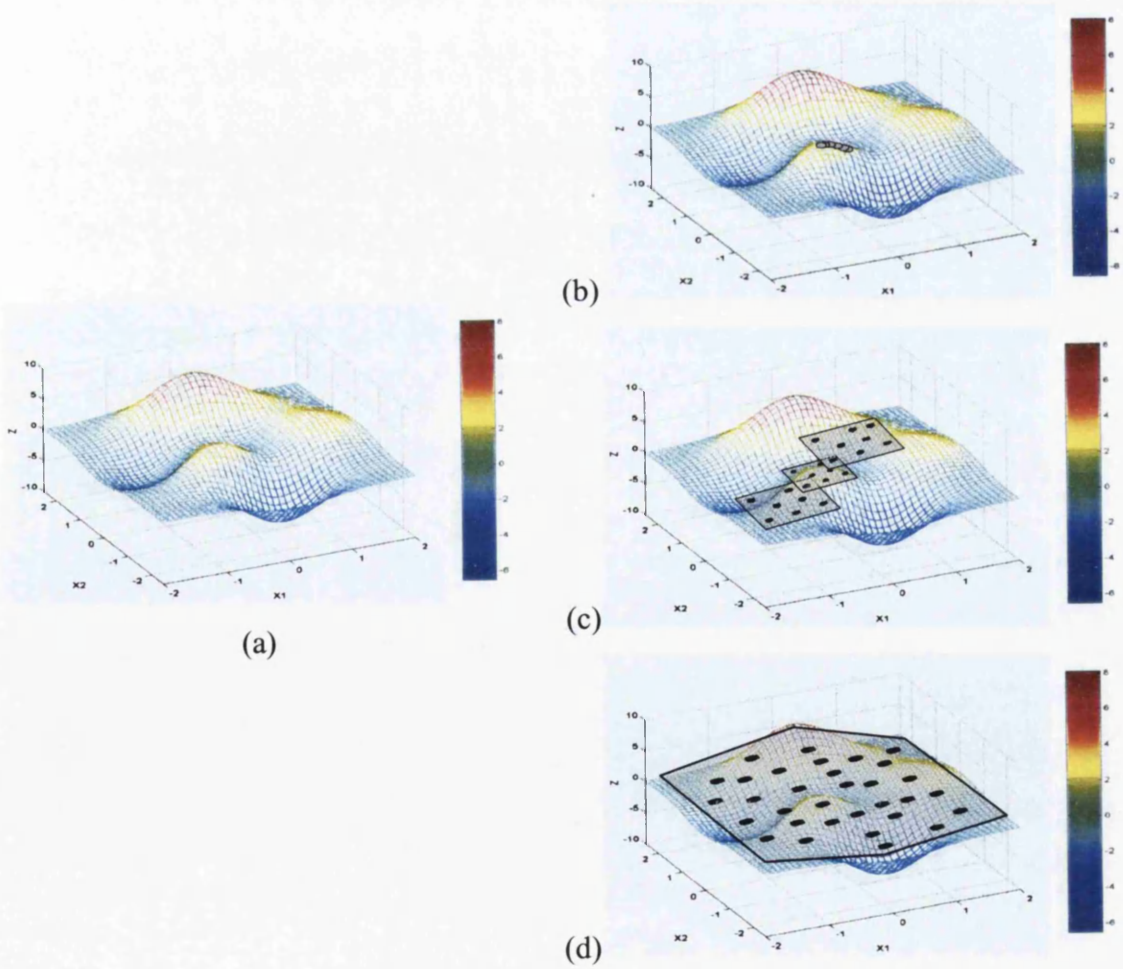


Figure 3.3.1: (a) Example design space; (b) local approximations; (c) mid-range approximations; (d) global approximations

Toropov [3.37] states, that the formulation of the approximate response surface is a result of balancing accuracy with simplicity. There is no point in producing a highly accurate function that is complex and contains high levels of noise. This defeats the object of the approximation, i.e. an approximation that is easy to evaluate and has low noise levels. The approximate response functions must be flexible enough to mimic the behaviour of the response functions with sufficient accuracy. On the other hand, they must be easy to evaluate and possess only a minor level of numerical noise, if any.

Barthelemy and Haftka [3.36] give a comprehensive review of the basic approximation techniques, other reviews include Sobieszczanski-Sobieski and Haftka [3.38], Jin *et al.* [3.39], Giunta [3.40] and Simpson *et al.* [3.41]. Simpson *et al.* [3.42] gives a complete discussion on the current issues with approximation methods.

This section gives an overview of the various design space approximation techniques including polynomial approximations (PA), which was used to generate the globally approximating response surfaces used in Chapter 6. The section is here for completeness as no specific development of the approximation techniques has been carried out in the work, although it has been appreciated that the use of approximation techniques is a very important aspect of design optimization.

3.3.1 Local approximations

Local approximations are valid in the region of the point at which they are generated. They are based on the "exact match of function values and their derivatives at one point of the design variable space using the first or second-order Taylor series."[3.43]. Local approximations can be used to reduce the complexity of the problem but convergence to local optima is likely and they don't address the issue of noise in the function value. Barthelemy and Haftka [3.36] and Alvarez [3.44] discuss in detail the local approximation methods. These methods include linear and reciprocal approximation, which, when combined, give the conservative approximation [3.44]. Alvarez [3.44] notes that the most popular local approximation techniques include CONLIN [3.22] and MMA [3.23]. Haftka [3.45] discusses a method for combining global and local approximations.

3.3.2 GLOBAL APPROXIMATIONS

Approximations of the entire design space are 'global approximations' and they are used to locate the region of the optimum solution. A variety of global approximation techniques are available: the response surface methodology (RSM) [3.46, 3.47] including polynomial regression and genetic programming [3.48, 3.44], artificial neural networks (ANN) [3.49, 3.50], and the design and analysis of computer experiments (DACE) which are based upon Kriging models, see Sacks *et al.* [3.51] and Booker *et al* [3.52].

3.3.2.1 RESPONSE SURFACE METHODOLOGY

RSM was originally developed by Box and Draper [3.47] to analyse experimental data and to create empirical models of the observed response values. "The particular strength of RSM is its applicability to investigations where there are few observations because the physical experiment is both very expensive and very time consuming to perform" [3.53] and in situations where there is noise in the response function.

RSM is used in design space approximation by fitting approximation functions to experimental (physical or virtual) data by the least-squares method. The choice of evaluation points is important to getting a good approximation of the response surface, especially when evaluations are expensive. The methodologies used for formulating the evaluation points are collectively known as *Design of experiments* (DoE). These are discussed in Section 3.4.

A variety of RSM techniques are available, the most popular is polynomial approximations, in particular, second-order models. Other methods include intrinsically linear approximations, mechanistic models, simplified numerical (low-fidelity) models and genetic programming models, all of which are discussed by Toropov and Alavrez [3.37, 3.54].

The following (taken from Ref. [3.37]) gives the basic requirements of a RSM and points out that the approximation need not necessarily be an explicit function. It could be an implicit one if some numerical procedure is involved in its formulation. The basic requirements to such a model can be summarized as:

- it must depend on the same design variables as the original function;
- it has to contain some tuning parameters to be defined using the general (non-linear) least-squares method;
- it must be simple enough to be used in numerous repeated calculations;
- it should not contain any considerable level of numerical noise in order not to cause convergence problems in the optimization process.

3.3.2.1.1 POLYNOMIAL APPROXIMATIONS

PA is often used for response surface modelling. Jin *et al.* [3.39] discusses PA and state that for problems with a large dimension, it is important to use linear or second-order polynomial models to narrow the design variables to the most critical ones. The authors also observe, that in optimization, its smoothing capability allows quick convergence of noisy functions. In spite of the advantages, there is a drawback when applying PA to model highly non-linear behaviours. Higher-order polynomials can be used; however, instabilities may arise, or it may be too difficult to take sufficient sample data to estimate all of the coefficients in the polynomial equation, particularly in with a high number of dimensions.

In Chapter 6, four second-order polynomial models are developed and used as metamodels in an optimization loop. The models are based on a general second-order polynomial model, which can be expressed as

$$y = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} a_{ii} x_i^2 + \sum_{i=1}^{n-1} \sum_{j=2}^{n} \underset{i<j}{a_{ij} x_i x_j}\,, \qquad (3.3.1)$$

where y is the response, x_i and x_j are the design variables and a are the tuning parameters, which are estimated using the least squares method. Therefore, determining the tuning parameters, using the response values at the DoE specified evaluation points, generates the second-order approximations. Hence, the four approximation functions given by Equations 6.2(a)-(d), take the same form as Equation 3.1, but differ by the values of the tuning parameters.

3.3.2.1.2 INTRINSICALLY LINEAR APPROXIMATIONS

(Note that the following is adapted from Ref. [3.37]) A simple choice of the structure of the approximations is an intrinsically linear (with respect to the tuning parameters) model, e.g. a linear, and a multiplicative model (Equations 3.3.2a and b respectively):

$$y = a_0 + \sum_{i=1}^{n} a_i x_i \, . \tag{3.3.2a}$$

$$y = a_0 \prod_{i=1}^{n} (x_i)^{a_i} \tag{3.3.2b}$$

The advantage of these approximation functions is that a relatively small number of tuning parameters a, is to be determined and the corresponding least squares problem is solved easily. Generally, the multiplicative approximations show better convergence characteristics. Higher order approximations, e.g. second-order polynomials, lead to a significantly better quality of approximation. However, these approximation functions require a considerably larger number of designs to be evaluated.

3.3.2.1.3 MECHANISTIC MODELS

In Ref. [3.47] Box and Draper introduced the empirical model building theory and showed, that by developing a mechanistic model, i.e. a model based upon the physics of the system being studied, the approximation function is better than one that is empirically based. "A difficulty in using such models is that they depend on the specific features of the problem and the researcher's experience."[3.37]

3.3.2.1.4 SIMPLIFIED NUMERICAL MODELS

A more general way of constructing high quality approximations is to obtain a simplified numerical or 'low-fidelity' model by simplifying the analysis model e.g. by using a coarser finite element mesh discretization or simpler geometry. The low-fidelity model contains the most important features of the high fidelity model but the simulation time is reduced. "In such a case a simplified numerical model provides a good basis for development of high quality approximations."[3.37]

3.3.2.1.5 Genetic Programming

During this research the area of genetic programming (GP) Koza [3.48] applied to design space approximation by Toropov and Alvarez [3.54, 3.37] and Alvarez [3.44], has been of noticeable interest. GP is a branch of GAs and, instead of using a string of numbers to represent the solution, the GP creates a random population of computer programs with a tree structure. For example, Equation 3.3.3 can be represented by the tree structure depicted in Figure 3.3.2(a), and the genetic operators of crossover and mutation are shown in Figure 3.3.2(b) and (c) respectively. It can be seen, that the programs consist of function and terminal nodes. The function nodes are mathematical operators such as $+$, $-$, $\times$, $/$, x^y, $\log x$, $\sin x$ etc. The terminal nodes are the design variables, $\underline{x}$.

$$\left(\frac{x_1}{x_2} + x_3\right)^2 \tag{3.3.3}$$

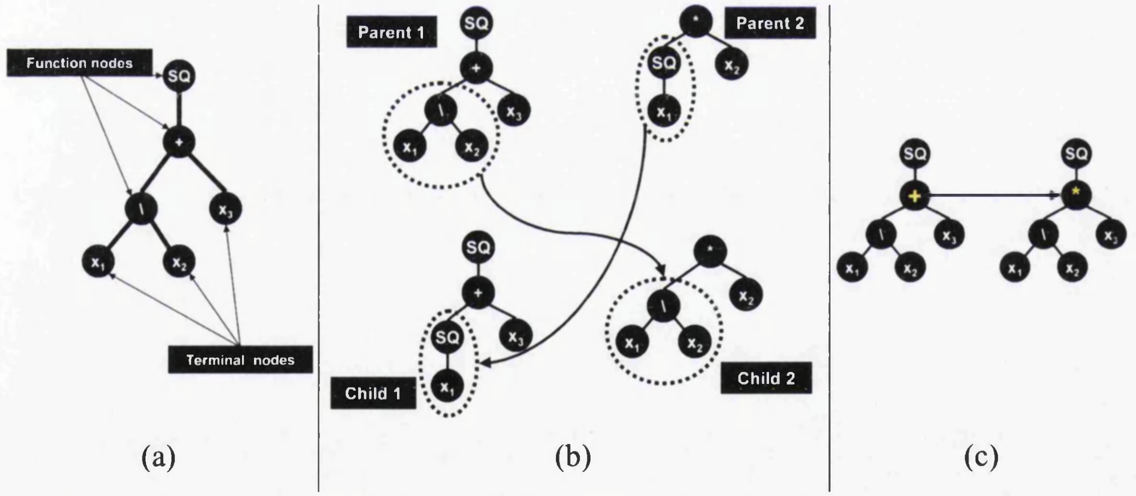


Figure 3.3.2: Genetic programming: (a) the tree structure of Equation 3.3.3; (b) an example of the crossover operator; (c) an example of the mutation operator

GP can be used to generate the approximation functions without prior knowledge of the structure of the function. The evolution of the programs is performed through the action of the genetic operators (reproduction, crossover, mutation and elite transfer). The evaluation of the fitness, which is usually calculated using the least-squares method, is a measure of the approximation quality.

3.3.2.2 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANN) are an alternative approach to global function approximation; they are analogous to biological nervous systems and adaptive biological learning. It consists of highly interconnected processing elements (neurons) that are tied together with weighted connections (synapses). Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. With ANN, learning typically occurs by iteratively adjusting the weights to fit a set of input/output data. The weights store the knowledge necessary to solve specific problems. In general, an ANN consists of an input layer followed by hidden layers and then an output layer. There are a number of ways of connecting the layers:

- Fully connected - each neuron on the first layer is connected to every neuron on the second layer.
- Partially connected - a neuron of the first layer does not have to be connected to all neurons on the second layer.
- Feed forward - the neurons on the first layer send their output to the neurons on the second layer, but they do not receive any input back form the neurons on the second layer.
- Bi-directional - there is another set of connections carrying the output of the neurons of the second layer into the neurons of the first layer.

A disadvantage of ANN is that the model is a black box, i.e. there is no analytical function to express the output for a given set of inputs. Therefore, no understanding of the underlying physical relationship can be identified.

Simpson *et al.* [3.41] give a description on the implementation of ANN. Lauridsen *et al.* [3.55], Carpenter and Barthelemy [3.56] compare the use of ANN to a polynomial response surface as do Dornberger *et al.* [3.34] who do the comparison by approximating the following function

$$\begin{aligned} f(x_1,x_2) = {} & 3(1-x_1)^2 \exp\left(-x_1{}^2-(x_2+1)^2\right) \\ & -10\left(\tfrac{1}{5}x_1 - x_1{}^3 - x_2{}^5\right)\exp\left(-x_1{}^2-x_2{}^2\right) \\ & -\tfrac{1}{3}\exp\left(-(x_1+1)^2-x_2{}^2\right) \end{aligned} \tag{3.3.4}$$

The design space for this function is shown in Figure 3.3.1(a). PA and ANN are developed to approximate this function at 200 randomly chosen sample points (red points in Figure 3.3.3). Figure 3.3.3 shows these two approximations and it can be seen, that ANN is a superior approximation of the design space with a maximum error of 1% and that the polynomial expression causes oscillations at the boundary. However, the training time to reach the ANN solution is 10 to 20 times longer than for the PA solution. The simulation time of ANN is five times greater than PA.

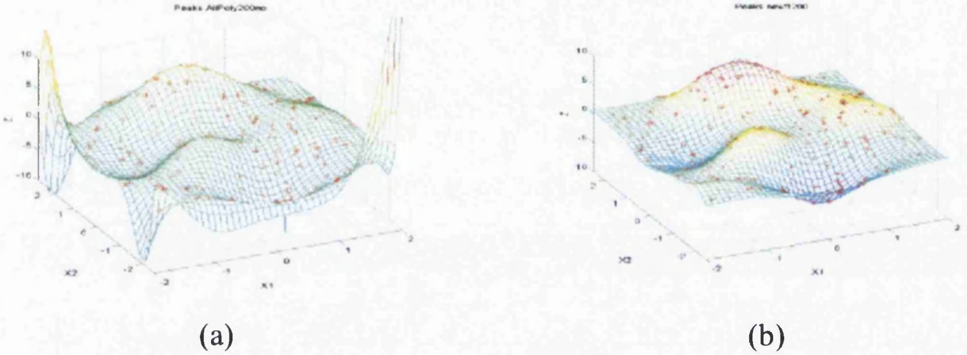


Figure 3.3.3: Approximations, generated in Ref. [3.34], of Equation 3.4 for (a) a polynomial model and (b) a neural network model

Sellar *et al.* [3.57] and Liu *et al.* [3.58, 3.59] developed an approach to develop response surface approximations based upon artificial neural networks incorporating gradient information into response surface approximations, thereby "reducing the number of function evaluations and improving the accuracy." [3.58] Other references include Smith [3.60], Cheng and Titterington [3.50].

3.3.2.3 DESIGN AND ANALYSIS OF COMPUTER EXPERIMENTS

"There is an important distinction between physical experiments, which have random error, and computer experiments, which are often deterministic." [3.42] Traditional response surface methods [3.47] assume that experiments have random errors, i.e. noise. DACE, as proposed by Sacks *et al.* [3.51], assumes that the responses are deterministic hence, there is no random error. The DACE approximations typically use an interpolation model to give an exact fit of the data. These models are called the Kriging function and are constructed using maximum likelihood estimation. The sample points are chosen by a space filling design, such as the optimal Latin Hypercube (Section 3.4), since the "classical ideas of replication and randomisation are irrelevant when it comes to deterministic computer experiments and very little is known *a priori* about the shape of the response function"[3.42].

For more details see, e.g., Sacks *et al.* [3.51], Booker *et al.* [3.52], Giunta and Watson [3.61], Simpson *et al.* [3.62, 3.63, 3.64], Martin and Simpson [3.65], Leary *et al.* [3.66] and Srivastava *et al.* [3.67].

3.3.3 MID-RANGE APPROXIMATIONS

Mid-range approximations are, as the name suggests, approximations of a region of the design space; they make use of the function and (possibly) gradient information obtained at two or more points unlike local approximations, which use data from a single point of the design space. Compared to global approximations "the accuracy requirements for mid-range approximations can be more relaxed, although faster convergence is possible using more accurate approximations."[3.37]

A mid-range approximation method called the 'multipoint approximation method based on the response surface fitting' (MARS) has been developed and used in a number of studies including [3.68, 3.69, 3.70, 3.71, 3.72], MARS is discussed in Section 2.3.4 and is applied in Chapter 5.

Other work on the use of mid-range approximations can be found in Schoofs *et al.* [3.73], Xu and Grandhi [3.74], Magazinović [3.75]. Venter [3.76] gives an extensive literature review.

3.4 DESIGN OF EXPERIMENTS

DoE is a strategy for planning experiments, which was introduced in the early 1920s by Fisher [3.77]. "Although modifying one variable at a time can sometimes lead to an improved design, this can be very inefficient and ineffective when interactions among design variables induce unforeseen results. The designer might try varying numerous inputs independently before one with significant impact is found, or might even resort to changing many variables simultaneously in a desperate attempt to evaluate the effect of a number of inputs at once. The former strategy cannot account for interaction effects, and the latter makes separation of any effects impossible when done in an informal/random manner."[3.78]

DoE methods are used to select the system parameter values and their combinations at which, experiments or simulations are to be run. The results of these experiments are then used to study the effect that changing the system parameters has on the system response, and in regression analysis to generate response surface models.

In this work DoE is used in several problems:

- In Chapter 5 DoE is used for defining the sample points, which are used in the optimization method, MARS (Chapter 2).
- In Chapter 5 DoE is used in a robust design optimization loop to define the sampling points used to assess the variation in the design performance caused by variation in the system parameters.
- In Chapter 6 DoE is used for defining the sample points that are used to develop quadratic response surface approximations of the design space.

There are a number of classical DoE methods such as Full-factorial, Fractional-factorial, Orthogonal arrays, D-optimal, Central-composite (CCD), Plackett-Burman, Box-Behnken (BBD). Myers and Montgomery in Ref. [3.46] give a comprehensive description of each of these methods. Space filling experimental designs include Latin Hypercube (LH) (McKay *et al.* [3.79]), uniform designs (Fang *et al.* [3.80]) and Hammersley sampling sequences (Kalagnanam and Diwekar [3.81]). In this work, three of these techniques have been used: CCD, BBD and LH methods. For completeness, the concept behind each of these methods is presented below in Section 3.4.1. Then in Section 3.4.2 a new technique to formulate the optimal LH DoE (OLH) is presented.

3.4.1 CONCEPT

In general, in order to generate a DoE, the upper and lower bounds for the system parameters or variables is firstly defined. Then, the range is discretized into equally spaced levels. In a two-level experiment the system parameters have two values at the extremities of their ranges. In a three-level experiment, the system parameters are at the two extremes and an additional level between these. For example, for a full factorial DoE with two levels, the total number of experiments is 2^n, where n is the number of variables. Similarly, a three-level full factorial design has 3^n experiments. The construction of a quadratic response surface model in n variables requires at least $(n+1)(n+2)/2$ response evaluations. With values of $n>10$ the 2^n and 3^n full factorial designs becomes impractical, for example for a two-level full factorial DoE with 10 variables $2^{10}=2048$ experiments are required.

A desirable property of a DoE used for response surface fitting is that the design is 'rotatable', so that the experimenter can have confidence that the quality of the response is the same throughout the design space. "A design is rotatable if the variance of the predicted response at any point **x** depends only on the distance of **x** from the design centre point. A design with this property can be rotated around its centre point without changing the prediction variance at **x**"[3.82].

3.4.1.1 CENTRAL-COMPOSITE DOE

CCD was introduced by Box and Wilson in Ref. [3.83] and can be used for fitting second-order response surface models. In CCD, a two-level full factorial experimental design is combined with $2n$ 'star' points and one or more centre points. The factorial points are used to estimate the interaction between the factors, the star and centre points are used to estimate the quadratic terms of the response surface and hence its curvature. The total number of experiments is 2^n+2n+1.

A two variable (x_1 and x_2) CCD is shown in Figure 3.4.1(c), which combines (a), the factorial points with (b), the star and centre points. In CCD, the star points lie outside the boundary created by the 2^n full factorial points. The distance from the star points to the centre of the CCD typically varies from 1.0 to $\sqrt{n}$. It can either be scaled so that they are the extreme values, this is known as 'inscribed', or the factorial points are the extremes and the star points lie beyond them, this is known as 'circumscribed'. Both of these variations are rotatable. As with 2^n and 3^n full factorial designs, the number of required CCD experiments also becomes impractical as n becomes large.

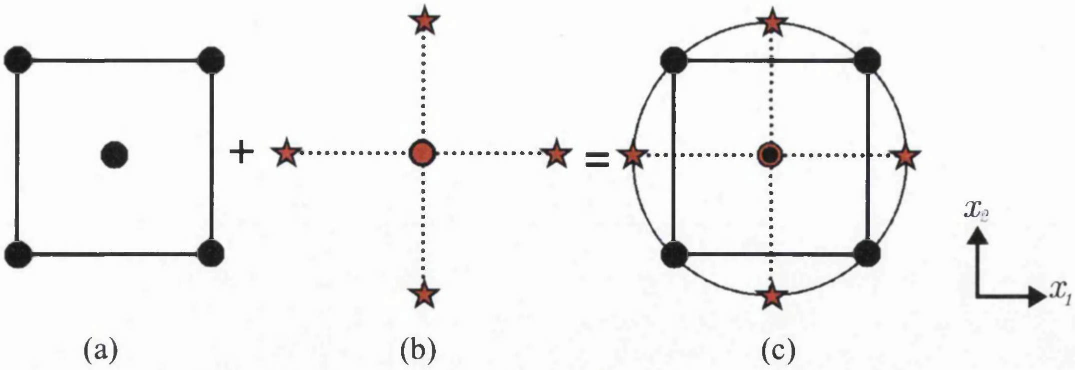


Figure 3.4.1: CCD for two variables x_1 and x_2; (a) factorial points, (b) star and centre points, (c) Combined factorial, centre and start points

3.4.1.2 Box-Behnken DoE

BBD designs were developed by Box and Behnken in Ref. [3.84]. In these designs, three levels are used for each factor and like CCD, can be used to fit a full quadratic model and they are rotatable. An example of a BBD is given in Figure 3.4.2 for two design variables, x_1 and x_2. It can be seen from Figure 3.4.2, that the corners of the square enclosing the design are not included in the design. So, it can be expected that these DoE have poorer prediction ability at the extreme combinations of the variables.

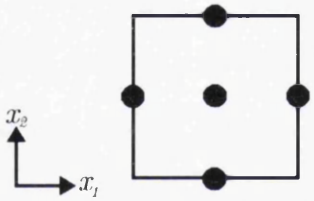
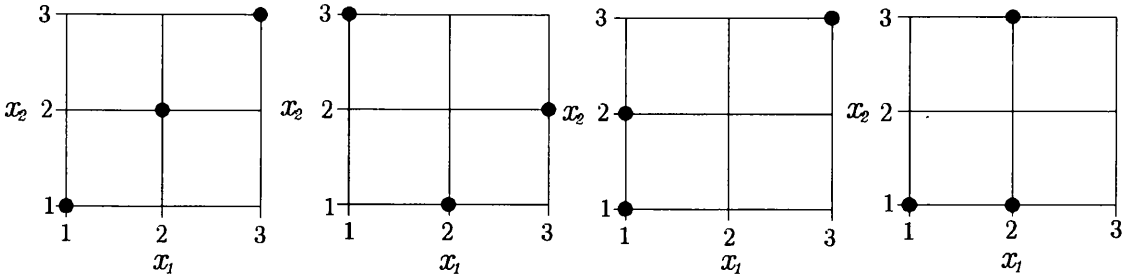


Figure 3.4.2: BBD for two variables x_1 and x_2

3.4.1.3 Latin Hypercube DoE

Both CCD and BBD are based upon developing a mathematical model of the process. LH, proposed by McKay et al. [3.79] and Iman and Conover [3.85], is independent of the mathematical model of a problem. It is structured so that each variable is divided into *P* equal levels. For each level, there is only one point (or experiment). For example, if *P*=3, i.e. three points, the range for each of the variables is the integer values 1–3. Figure 3.4.3 shows some permissible and impermissible combinations for two design variables (*N*=2) and three points (*P*=3).



(a) permissible (b) permissible (c) impermissible (d) impermissible

Figure 3.4.3: Permissible and impermissible LH DoE for two design variables, *N*=2 and three points, *P*=3

Two LH methods are the random sampling LH method (RLH) and the optimal Latin Hypercube designs (OLH). RLH and OLH differ by how the points in the DoE are distributed. The RLH method uses random sampling to get each point in the DoE, whereas the OLH methods use more structured approaches with the aim of optimizing the uniformity of the distribution of the points. The generation of the OLH DoE is time consuming, e.g. a DoE with 10 points and 5 design variables has 6×10^{32} possible solutions. If each solution takes one nanosecond (1×10^{-9} seconds) to evaluate, it takes 2×10^{16} years to find the optimum by enumeration. This is clearly infeasible and therefore solving this minimization problem requires a more advanced optimization technique to search the design space.

It is important to note that the LH DoE for N variables and P points is independent of the application under consideration. Once the DoE for N variables and P points is formulated, re-calculation of the DoE is not required. The DoE matrix for N variables and P points in Figure 3.4.4 shows the DoE for N=3 and P=4. The matrix is scaled to fit any range of the design variables. Therefore a LH DoE for a problem with N=3 and P=4 is generally determined by this matrix.

For example, if: x_1 ranges from 5.0 to 100.0, x_2 from -2.0 to 3.0 and x_3 from 10.5 to 12.0. Then, using the scaling matrix in Figure 3.4.4:

- point 1 = (5.0, −2.0, 11.0),
- point 2 = (36.67, 3.0, 11.5),
- point 3 = (100.0, −0.333, 12.0),
- point 4 = (68.333, 1.333, 10.5).

Point	x_1	x_2	x_3
1	1	1	2
2	2	4	3
3	4	2	4
4	3	3	1

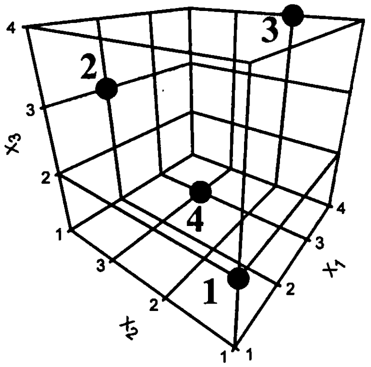


Figure 3.4.4: LH DoE for three variables $N = 3$ and four points $P = 4$

3.4.2 OPTIMAL LATIN HYPERCUBE DESIGN OF EXPERIMENTS

In Chapters 4 and 5 the development and implementation of a robust design procedure is described. Part of the robust design optimization loop requires a space-filling DoE such as the OLH. From the literature it is evident that a general procedure for formulating the OLH has not been fully established and there is a need for a procedure to be developed and implemented. This section describes two new methods, developed in this work, for formulating the OLH. The reader is referred to Section 3.4.1.3 for a brief overview of LH and the requirement of optimization to generate OLH.

3.4.2.1 INTRODUCTION

Several methods have been proposed to generate OLH using criteria such as maximizing entropy [3.86], integrated mean-squared error [3.87], and the maximization of the minimum distance between points [3.88]. Audze and Eglais [3.89] proposed a method (abbreviated here as AELH) that uses the potential energy of the points in the DoE to generate a uniform distribution of points. Jin *et al.* [3.90] introduce an 'enhanced stochastic evolutionary algorithm' for formulating OLH.

This work introduces two methods for formulating the OLH DoE by optimization. Firstly, a method for generating OLH using a binary encoded GA (binGA) [3.91, 3.92, 3.93] is described, and then secondly, using a permutation GA (permGA) [3.94, 3.95]. The methods are independent of the objective function and they are applied here to generate OLH using the AELH formulation described in Section 3.4.2.2. The results are compared to RLH DoE from [3.96] and existing tabulated results in [3.97].

The work highlights the shortcomings of the binGA for the solution of the OLH and the considerable improvements gained using the permGA method; permGA generates better solutions than binGA, and the computational effort in reaching those solutions is significantly reduced.

3.4.2.2 AUDZE-EGLAIS OBJECTIVE FUNCTION

The Audze-Eglais method is based on the following physical analogy: a system consisting of points of unit mass exert repulsive forces on each other causing the system to have potential energy. When the points are released from an initial state, they move. They will reach equilibrium when the potential energy of the repulsive forces between the masses is at a minimum. If the magnitude of the repulsive forces is inversely proportional to the distance squared between the points then minimizing Equation (3.4.1) will produce a system of points distributed as uniformly as possible

$$\min U = \min \sum_{p=1}^{P} \sum_{q=p+1}^{P} \frac{1}{L_{pq}^{2}} \qquad (3.4.1)$$

where U is the potential energy and L_{pq} is the distance between the points p and q $(p \neq q)$. For two design variables (N=2) and three points (P=3) the design of experiments shown in Figure 3.4.5 is one possible solution to the AELH DoE. The quality of the solution is calculated using the objective function in Equation (3.4.1). The solution in Figure 3.4.5 consists of point 1, located at (1,1), point 2, located at (2,2) and point 3, located at (3,3).

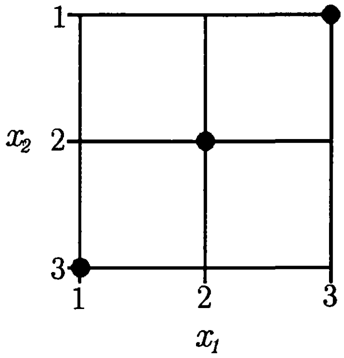


Figure 3.4.5: DoE for two variables (*N*=2) and three points (*P*=3)

Various DoE combinations can be evaluated and the DoE with the minimum objective function is the AELH DoE. The search for the best DoE, found by minimizing Equation

(3.4.1), can be carried out either by optimization or by screening every possible combination. By the argument given in Section 3.4.1.3 and from Table 3.4.1, the screening approach is virtually infeasible since the search through every possible combination is cumbersome and requires excessive computation time.

Points (P)	$N = 2$		N=5	
	Possible positions for first point	Total number of combinations	Possible positions for first point	Total number of combinations
1	1	1	1	1
2	4	4	32	32
3	9	36	243	7776
4	16	576	1024	7,962,624
5	25	14,400	3125	24,883,200,000
10	100	1.32×10^{13}	100,000	6.29×10^{32}
30	900	7.04×10^{64}	2.43×10^{7}	1.31×10^{162}

Table 3.4.1: The number of possible positions for the first point in the DoE and the total number of combinations. Where *P* is the total number of points and *N* is the number of design variables in the DoE

3.4.2.3 Optimization using a binary encoded genetic algorithm

To meet the requirement of one point for each level (described in Section 3.4.2.2) the objective function (Equation (3.4.1)) is modified to penalize DoE with more than one point in a level. The initial population of the GA is generated such that no individuals violate the rule of no more than in point in each level. However, the genetic operators of mutation and crossover may cause an individual to break this rule and so a penalization method is required. Section 3.4.2.3.1 describes the penalization method used.

Optimization also requires the design variables of the problem to be encoded. The design variables for this problem are the points of the DoE, so encoding of these points into a form understood by the optimizer is required. Two possible encoding methods use:

1. Node numbers (described in Section 3.4.2.3.2)
2. Co-ordinates (described in Section 3.4.2.3.3)

It is important to note the following definitions:

- *P* is the number of points in the DoE.
- *N* is the number of design variables in the problem for which the DoE is being formulated.
- *K* is the number of design variables used to represent the DoE.

3.4.2.3.1 PENALIZATION METHOD

The penalization method used in this study counts the number of points in the same level and increases the objective function accordingly so that the modified objective function, F is:

$$\left| \begin{array}{l} \text{if } P_r = 0: \; F = U \\ \text{if } P_r > 0: \; F = 3 \times P_r \times U \end{array} \right. \tag{3.4.2}$$

where U is the potential energy calculated by Equation (3.4.1), P_r is the number of points in the same level and 3 is a constant chosen such that the function will always be greater if there are points in the same level than if there were not. In this study, the fittest individual i, has the lowest potential energy. Therefore the fitness function is:

$$f_i = \left(F_{\max} + F_{\min}\right) - F_i \tag{3.4.3}$$

where f_i is the fitness of an individual i, $F_{\max}$, $F_{\min}$ and F_i are the maximum, the minimum and the individual modified objective function values respectively. The GA works by searching the design space for the individual with the highest fitness (calculated by Equation (3.4.3)), hence minimizing Equation (3.4.2).

3.4.2.3.2 ENCODING USING NODE NUMBERS

This approach encodes each point in the DoE by a single node number akin to finite FEs. So for example, Figure 3.4.5 becomes Figure 3.4.6 and the encoding of this solution changes from (1,1) (2,2) (3,3) to $\boxed{1}\ \boxed{5}\ \boxed{9}$. The encoding can then be used in the optimiser with K points where K is equal to P design variables (as defined in Equation (3.4.4)).

$$K = P \tag{3.4.4}$$

For example, if P=120 and N=5 then K=120. So, 120 design variables are used, each being an integer between 1 and the number of nodes. In this case the number of nodes is 120^5(=24,883,200,000) which, using Equation (3.4.5) with $n = 120^5$, requires a bit length of 35. This method requires converting the node numbers back to co-ordinates when calculating the potential energy of the system:

$$n = 2^L \qquad L = \frac{\log n}{\log 2} \tag{3.4.5}$$

where L is the bit length and n is the maximum design variable value.

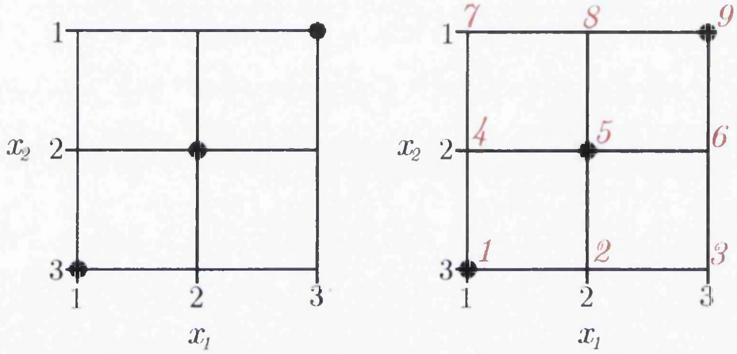


Figure 3.4.6: Original (left) and node numbers based encoding (right)

3.4.2.3.3 Encoding using co-ordinates

This approach encodes the co-ordinates of each point. The first P design variables being the x_1 co-ordinates, the next P design variables being the x_2 co-ordinates etc. up to N variables. So the encoding of Figure 3.4.6 (left), i.e. (1,1) (2,2) (3,3) becomes $\boxed{1}\,\boxed{2}\,\boxed{3}\,\boxed{1}\,\boxed{2}\,\boxed{3}$. The encoding is then used in the optimizer with the number of design variables K calculated by Equation (3.4.6a), meaning that each co-ordinate of each point becomes a design variable:

$$K = PN \tag{3.4.6a}$$

K can be reduced by making the x_1 co-ordinates fixed from 1 to the number of points and so the equation for K becomes:

$$K = P(N-1) \tag{3.4.6b}$$

For example, if P = 120 and N = 5 then 480 design variables are used. The x_1 co-ordinates are fixed from 1 to 120, and the other co-ordinates are the design variables each being an integer between 1 and 120. So in Equation (3.4.5), n = 120 and the corresponding bit length is 7.

3.4.2.3.4 WHICH ENCODING METHOD?

The decision as to which encoding method to use is based on the chromosome length (the product of the bit length and the number of design variables) of an individual in a population, and numerical limitations. The lower the chromosome length the less work the GA has to do in locating the optimum. The lower the value of the maximum design variable, the lower the chance of numerical errors.

Table 3.2 lists, for various values of P and N, the maximum values of the K design variables and the chromosome lengths for the two methods.

P	*N*	Maximum value of design variables		Bit length		Chromosome length	
		coordinates method	nodes method	coordinates method	nodes method	coordinates method	nodes method
2	2	2	4	1	2	2	4
2	3	2	8	1	3	4	6
2	4	2	16	1	4	6	8
2	5	2	32	1	5	8	10
10	2	10	100	4	7	40	70
10	3	10	1000	4	10	80	100
10	4	10	10,000	4	14	120	140
10	5	10	100,000	4	17	160	170
50	2	50	2500	6	12	300	600
50	3	50	125,000	6	17	600	850
50	4	50	6,250,000	6	23	900	1150
50	5	50	312,500,000	6	29	1200	1450
120	2	120	14,400	7	14	840	1680
120	3	120	1,728,000	7	21	1680	2520
120	4	120	207,360,000	7	28	2520	3360
120	5	120	24,883,200,000	7	35	3360	4200

Table 3.2: List of bit lengths, chromosome lengths and maximum values of the *K* design variable values for various values of *P* and *N*, where *P* is the number of points in the DoE, *N* is the number of design variables in the problem for which the DoE is being formulated, and *K* is the number of design variables used to represent the DoE

Using Table 3.2, for a DoE with 120 points and 5 design variables, comparison of the chromosome lengths for the two methods is as follows. The 'node numbers' method uses 120 design variables (Equation (3.4.4)), each requiring a bit length of 35 (Equation (3.4.5)), this gives a chromosome length of 4200 (35×120). The 'co-ordinates' method uses 480 design variables (Equation (3.4.6b)) each requiring a bit length of 7. This gives a chromosome length of 3360 (7×480). In this example, the 'co-ordinates' method requires 20% fewer bits for encoding the chromosomes and uses much smaller design variables values, hence it has a lower risk of running into numerical problems.

Based on the above argument the 'co-ordinates' method is the general method used for this study.

3.4.2.4 OPTIMIZATION USING A PERMUTATION GENETIC ALGORITHM

The requirement of one point in each level for OLH is similar to the travelling salesman problem (TSP): the travelling salesman must visit every city in the area exactly once and return to the starting point. Given the cost of travel between all cities, how should the itinerary be planned such that the cost of the entire tour is minimized? The only difference between the TSP and the formulation of the LH is that in the LH problem the 'salesman' doesn't return to the starting point. An extensive overview of many approaches to solving this problem is given in Ref. [3.94]. One method is to use a GA to find the optimal 'permutation' of cities where the term permutation, is "the rearrangement of existent elements, or the process of changing the lineal order of an ordered set of objects"[3.98].

The problem with using the binGA is that the crossover and mutation operators produce infeasible designs for the population. These solutions need to be penalized to guide the binGA towards feasible designs. Using permGA the encoding is done with integer values instead of binary units. Furthermore, the mutation and crossover operators are modelled such that the rule of one point in each level for the AELH is never contravened. This means that the optimization problem is unconstrained and no penalty factor is required. Therefore, using a method such as permGA will allow the GA to fully concentrate on the optimization without having to waste computational effort gradually filtering out infeasible solutions from the population.

3.4.2.4.1 ENCODING

The encoding of the AELH is done using the coordinates method described in Section 3.4.2.3.3. The first *P* design variables being the x_1 coordinates, the next *P* design variables being the x_2 coordinates etc. up to *N* variables. So from Figure 3.4.3(b), where the coordinates of the DoE points are (1,3) (2,1) (3,2), the encoding becomes $\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{3}$ $\boxed{1}$ $\boxed{2}$. In Section 3.4.2.3.3 this 'string' is converted to binary to give the chromosome representing this DoE as $\boxed{01}$ $\boxed{10}$ $\boxed{11}$ $\boxed{11}$ $\boxed{01}$ $\boxed{10}$. Using the permGA [3.94, 3.95] the encoding requires no conversion so that the string representing the DoE is $\boxed{1}$ $\boxed{2}$ $\boxed{3}$ $\boxed{3}$ $\boxed{1}$ $\boxed{2}$. The formulation here is that the first *P* numbers are a random sequence of numbers between 1 and *P*, the next *P* numbers are a random sequence of numbers between 1 and *P*. This is repeated up to *N*. There are no repetitions of numbers in each sequence therefore the rule of one point in each level is not contravened. Below is an example of using the genetic operators with a permutation encoding.

3.4.2.4.2 GENETIC OPERATORS

Mutation - two numbers are selected and exchanged e.g. 2nd and 5th

[4 **1** 5 2 **3**] ⇒ [4 **3** 5 2 **1**]

Crossover can be done in a variety of ways and is applied to each sequence of *P* numbers for the *N* variables. Three crossover methods have been implemented in this work: a 'simple crossover' method, the 'cycle crossover' method by Oliver et al. [3.99], and an 'inversion' method.

Simple crossover

A crossover point is selected (2 in this example), the permutation is copied from the first parent until the crossover point, then, starting from the beginning, the other parent is scanned and if the number is not yet in the child, it is added.

Parent 1 = [**4 1** 3 5 2] ⇒ Child 1 = [**4 1** 5 2 3]

+

Parent 2 = [**5 2** 1 4 3] ⇒ Child 2 = [**5 2** 4 1 3]

Cycle crossover

In this method, each value and its position comes from one of the parents. The method preserves the absolute position of the elements in the parent sequence. An example of the implementation of cycle crossover is as follows (adapted from Ref. [3.94]):

Parent 1 = [1 3 9 7 5 4 6 2 8]
+
Parent 2 = [4 6 2 1 7 8 9 3 5]

Produces child 1 by taking the first value from the first parent:

Child 1 = [1 * * * * * * * *]

The next point is the value below 1 in parent 2, i.e. 4. In parent 1 this value is at position '6', thus

Child 1 = [1 * * * * 4 * * *];

This, in turn, implies a value of 8, as the value from parent 2 below the selected value 4. Thus

Child 1 = [1 * * * * 4 * * 8];

Following this rule, the next values in child 1 are 5 and 7. The selection of 7 requires the selection of 1 in parent 1, which is already used, meaning that a cycle is completed

Child 1 = [1 * * 7 5 4 * * 8].

The remaining values are filled from the other parent:

Child 1 = [1 6 2 7 5 4 9 3 8].

Similarly,

Child 2 = [4 3 9 1 7 8 6 2 5].

Inversion

In this method two cut-off points are chosen at random in a parent and the values between these points are inverted e.g. for the cut-off points 3 and 7 marked as '|'

Parent 1 = [1 3 9 | 7 5 4 6 | 2 8] ⇒ Child 1 = [1 3 9 | 6 4 5 7 | 2 8]

Other crossover methods include 'partially mapped crossover' by Goldberg and Lingle [3.100], and 'order crossover' by Davis [3.101] see Ref. [3.94] for further details.

3.4.2.5 Problem definitions

Eight OLH problems have been attempted in this study. The problems are defined by number of design variables and the number of points in a DoE and are as follows

Problem number	Number of variables × Points
1	2 × 5
2	2 × 10
3	2 × 120
4	3 × 5
5	3 × 10
6	3 × 120
7	5 × 50
8	5 × 120

Table 3.3: Attempted OLH problems

The binGA uses a combination of elitist and tournament selection and is set-up for each problem as shown in Table 3.4, using experience and trial-and-error:

Design	**Problem number** Number of variables × Points							
	1 2 × 5	**2** 2 × 10	**3** 2× 120	**4** 3 × 5	**5** 3 × 10	**6** 3 × 120	**7** 5 × 50	**8** 5 × 120
Population size	5	20	100	20	20	20	200	200
Crossover	1-point	1-point	1-point	1-point	1-point	1-point	1-point	1-point
Mutation	1-point	1-point	1-point	1-point	1-point	1-point	1-point	1-point

Table 3.4: binGA set-up

During the development of the permGA algorithm it has been found, that both the cycle crossover and the inversion work quite well when they are applied individually. It has been found however, that when the two are combined, first by applying the cycle crossover followed by inversion, the performance considerably improves. The solutions are good for each problem using the same set-up, i.e. a population size of 10, with a tournament size of 2, 1-point mutation and 10% elitist size.

3.4.2.6 Results

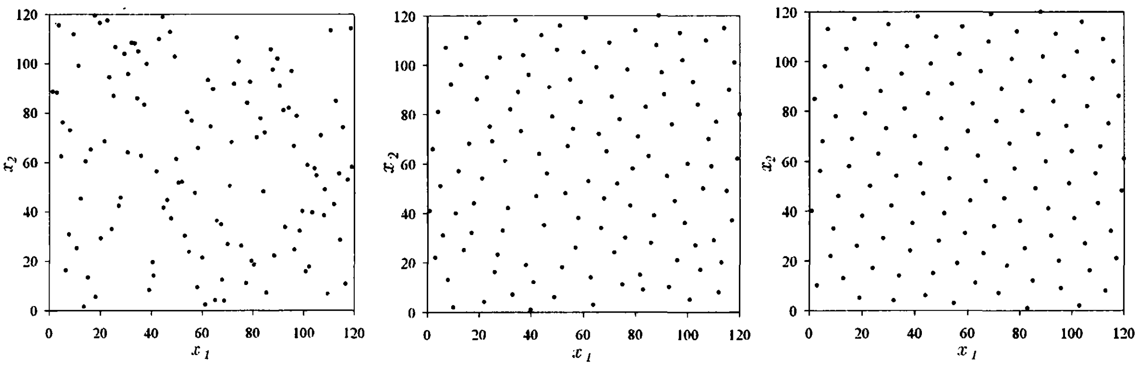
The results from the binGA, the permGA, the corresponding RLH DoE, and two existing DoE formulations from Ref. [3.97] are given in Table 3.5. Also shown in Table

3.4 is the number of function evaluations required to reach the binGA and permGA solutions. Comparisons with two previous DoE techniques from Ref. [3.96] and Ref. [3.97] show that the DoE formulated using the binGA and permGA methods are better because they have lower potential energies. It can be seen that the permGA solutions result in improvements over the binGA solutions and require on average 46-times fewer function evaluations to reach. It can be concluded, that the two methods work and that the permGA method is an effective tool for developing OLH DoE.

Design	**Problem number** Number of variables × Points							
	1 2 × 5	**2** 2 × 10	**3** 2 × 120	**4** 3 × 5	**5** 3 × 10	**6** 3 × 120	**7** 5 × 50	**8** 5 × 120
RLH from Ref. [3.96]	2.7471	4.0772	10.4438	2.1068	2.3020	2.6404	0.8849	0.8903
AELH –results from Ref. [3.97]	—	2.1065	—	—	—	—	0.7320	—
binGA	1.2982 (60)	2.0662 (39,240)	5.7733 (22,003,500)	0.7267 (5260)	1.0401 (165,980)	2.0541 (5,908,540)	0.7348 (280,000,000)	0.8003 (59,802,200)
PermGA	1.2982 (50)	2. 0662 (1860)	5.5174 (130,570)	0.7267 (1922)	1.0242 (38,950)	1.9613 (1,996,920)	0.7270 (1,996,840)	0.7930 (1,998,540)

Table 3.5: Potential energy comparison of RLH designs from Ref. [3.96], AELH designs from Ref. [3.97], binGA designs and permGA designs, with the number of function evaluations required shown in brackets where appropriate

The DoE for problem 3 (2 design variables and 120 points) is shown in Figure 3.4.7. Figure 3.4.7 (a) shows the RLH DoE generated using Ref. [3.96], (b) shows the OLH DoE formulated using the binGA and (c) shows the OLH DoE formulated using the permGA. It would be expected from the results given in Table 3.5 that, RLH has the worst uniformity and permGA has the best uniformity. Inspection of Figure 3.4.7 confirms this. Other results are given in Appendix A1.



(a) RLH (b) binGA (c) permGA

Figure 3.4.7: Plan points for problem 3 (*N*=2 and *P*=120), generated using several methods

3.5 MULTI-OBJECTIVE OPTIMIZATION

3.5.1 BACKGROUND

Many design optimization problems are 'multi-objective' (MO) in that they have more than one performance characteristic or objective. For example the design of a bridge might have two objectives: (i) minimizing its weight and (ii) maximizing its stiffness. A change in the design may either improve or worsen the solution depending on which criteria the design is judged. A design that is optimal for all the objectives is not usually possible, e.g. it is unlikely that the minimum weight bridge also has the maximum stiffness. Instead, there are many solutions to the problem, each representing a compromise between the objectives.

Pareto [3.102], an economist and sociologist, established the optimality concept in the field of economics based on multiple objectives. A Pareto optimal design is one where no other design is concurrently better in all objectives. A Pareto front contains all of the solutions representing the trade-off between competing objectives. Figure 3.5.1 highlights this concept, showing Pareto solutions and the inferior, discarded solutions. "It is generally accepted that a solution to a multi-objective optimization problem is Pareto optimal."[3.103]

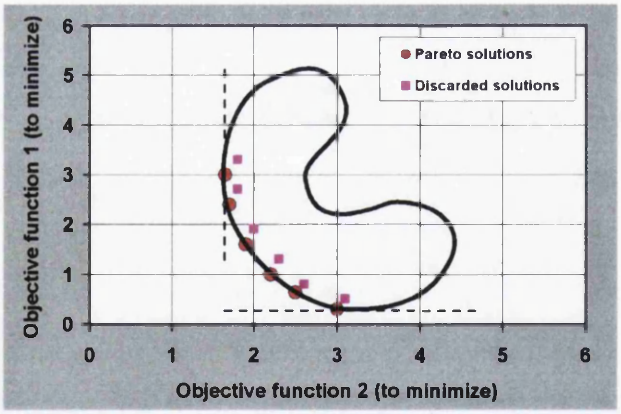


Figure 3.5.1 Pareto-optimization for multi-objective optimization

There are a number of methods for solving multi-objective problems. Such as the weighted sums method (WS), the normalized normal-constraint method (NNC), and multi-objective genetic algorithms (MOGA). These methods are summarized in the remainder of Section 3.5.1. For other techniques see Chen *et al.* [3.104]. The robust design framework developed in Chapters 4 and 5 requires a MO algorithm. In this work, two proven robust and efficient methods for solving MO problems have been implemented; these are the normal boundary intersection method (NBI) and the physical programming method (PP). Furthermore, a new MO method has also been developed, based upon PP, called the Pareto-front marching method (PFM-PP). These three techniques are discussed in detail in Section 3.5.2. Using the two-bar truss problem of Kirsch [3.105], a performance comparison is made between the current implementations of the methods, together with the results given in Ref. [3.106], generated using MOGA, this is described in Section 3.5.5.

Further comparison of the three methods, NBI, PP and PFM-PP is made using the bi-objective, 'robust design optimization', slit die design problem in Chapter 5.

3.5.1.1 WEIGHTED SUMS METHOD (WS)

This is a standard technique for generating the Pareto set in MO problems. The idea of the method is to minimize the sum of the objectives, which are weighted, and repeat this for various different weightings. The corresponding weighted-sum problem is

$$\sum_{i=1}^{n} w_i F_i(x), \tag{3.5.1}$$

where n is the number of weights, $w_i \geq 0$,$\sum_{i=1}^{n} w_i = 1$ and w_i are the weights assigned to the objective function $F_i(x)$. Say, for example, in a bi-objective problem, nine uniformly distributed Pareto points are desired. For the first point, a weighting of 0.1 is applied to objective 1 and a weighting of 0.9 to objective 2. The overall objective function is the sum of these 'weighted' objectives, i.e. $(0.1 \times \text{objective } 1)$ + $(0.9 \times \text{objective } 2)$, and the Pareto point is found using optimization to minimize this function. This procedure is repeated for the remaining eight points using different weight settings. Ideally, it may be assumed that a uniform distribution of points can be generated, using an evenly distributed set of weights. However, "it is a frequent observation that even for convex Pareto curves"[3.107] this idealization does not occur, even though, the method is able to get points from all parts of the Pareto set when the Pareto curve is convex. The reason behind this is "that, the weight is related to the slope of the Pareto curve in the objective space in a way such that, an even spread of Pareto points actually corresponds to often very uneven distributions of weights" [3.107]. In general, in order to get a uniform distribution of Pareto points using WS, the weightings have to be chosen using an iterative procedure, which is undesirable. "In the typical case where competing objectives are nonlinearly dependent on the design variables, iteratively guessing the appropriate values of the weights is - at best - a poor and inefficient way to engage in the design of complex systems." [3.108]

3.5.1.2 NORMALIZED NORMAL-CONSTRAINT METHOD (NNC)

This method was introduced by Messac *et al.* [3.103] and is an improvement over the normal constraint method by Yahaya and Messac [3.109] by removing numerical scaling problems through the normalization of the objectives. The method has been shown to be able to generate uniform spread of Pareto points. NNC works in a similar manner to the normal boundary intersection method (discussed later) and its graphical representation, taken from Ref. [3.103], can be seen in Figure 3.5.2.

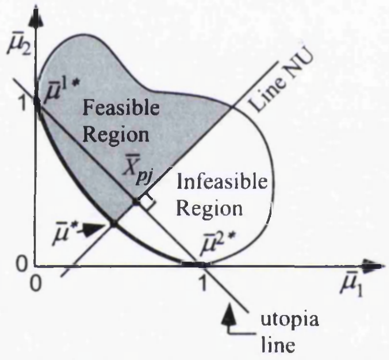


Figure 3.5.2: Graphical representation of the normal constraint method for bi-objective problems

The NNC procedure, for a bi-objective problem, is as follows:

(i) Using the notation of Ref. [3.103], the normalized utopia points, $\bar{\mu}^{1*}$ and $\bar{\mu}^{2*}$, are found by minimizing the individual objectives, $\bar{\mu}^{1}$ and $\bar{\mu}^{2}$, separately.

(ii) A line (or hyperplane for more than two objectives) is constructed between the utopia points. This is called the 'utopia line' and is equivalent to the 'convex hull of individual minima' used in the NBI method.

(iii) The utopia line is then divided into *n* equally spaced points (or weightings) depending on the number of desired Pareto points.

(iv) Normals to the utopia line are constructed through each of the *n* points. These normals then represent the 'normal constraint' in that, solutions below the line are considered to be infeasible, and solutions above the line are considered to be feasible. So, in Figure 3.5.2 the line NU represents a normal constraint, and the corresponding feasible and infeasible regions can be seen.

(v) Optimization is carried out by minimizing $\bar{\mu}^2$ subject to the normal constraint.

3.5.1.3 MULTI-OBJECTIVE GENETIC ALGORITHMS

The multi-objective genetic algorithm (MOGA) approach was proposed by Fonseca and Fleming [3.110]. The fitness assignment is rank based so that all non-dominated individuals are assigned rank 1 and the remaining are assigned a rank according to how many other individuals dominate them. Some current research in that area was carried out by Farhang-Mehr and Azarm [3.106] who developed an entropy-based MOGA (or E-MOGA), with the idea of maximizing the entropy of the Pareto front; compared to a baseline MOGA [3.111] the method produces a more uniform Pareto front for the same number of function calls. One example used in that paper was the two-truss problem of Kirsch [3.105], which is used to test the implementations of the following MO methods, the results of which are briefly described in Section 3.5.5.

3.5.2 NORMAL BOUNDARY INTERSECTION METHOD

The Normal Boundary Intersection method (NBI) for generating points on the Pareto surface of a multi-objective optimization problem was introduced in [3.112] by Das and Dennis. The main advantage of NBI is that the Pareto points are spread evenly on the Pareto surface. NBI was enhanced in [3.113] by Das with the introduction of the 'recursive knee approach' to address the issue that 'more Pareto points should be obtained in the more non-linear parts of the Pareto surface, and less in the relatively linear parts' [3.113]. The concept of NBI is described using the following example, which is a bi-objective problem. Subsequently, a formal description of the theory and the implementation used in this work is given.

Step 1: Minimize the objectives independently to identify the utopia points, $F(x_1^*)$ and $F(x_2^*)$. Next a straight line between these two points is drawn, this line is called the convex hull of individual minima (CHIM). Then, the CHIM is divided into k equally distributed weightings depending upon the number of points k required (k=30 in this example). Figure 3.5.3 shows the utopia points, the corresponding CHIM, and the k weightings.

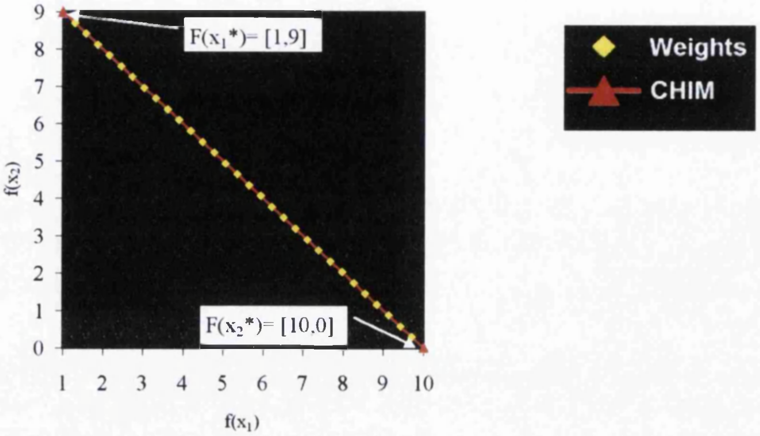


Figure 3.5.3: NBI method; utopia points and the CHIM divided into k weightings

Step 2: k uniformly distributed normals to the CHIM are constructed. The objective is to find the combinations of the design variables that maximize the distances along each of the normals. So for example, in Figure 3.5.4 for the current point i = 17 the weighting, β_{17} is calculated using the following equation for point i:

$$\beta_i = \frac{i}{k+1}, \tag{3.5.2}$$

where $\{\beta \in \Re^k, \sum_{i=1}^{k} \beta_i = 1, \beta_i \geq 0\}$. So, for i = 17 and k=30,

$$\beta_{17} = \frac{17}{30+1} = 0.548.$$

The objective of NBI is to maximize the distance (directed towards the origin) along the normal at i = 17, i.e. maximize distance t_{17} in Figure 3.5.4.

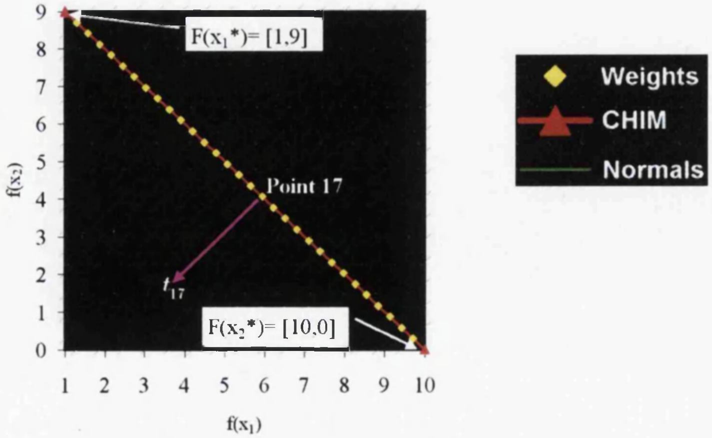


Figure 3.5.4: NBI method: calculate the k normals to the CHIM and find the maximum distance along each, e.g. point 17

Step 3: Repeat Step 2 k-times and find the maximum distance along each of the k normals (directed towards the origin). A typical results is shown in Figure 3.5.5 for k =30.

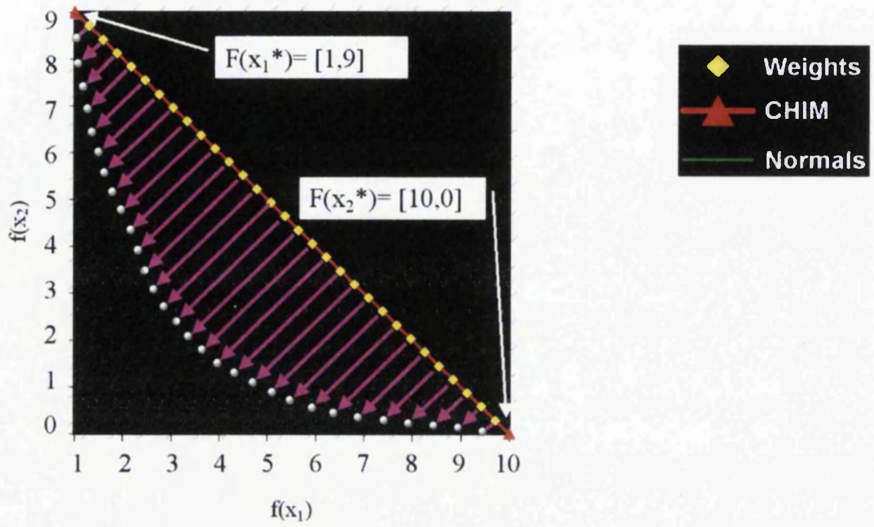


Figure 3.5.5: NBI method: calculate the maximum distance along each normal

To generate the k points along the Pareto front, the distance t_i is maximized for each of the k points. This requires k sub-optimization problems to be solved. In a more formal manner the method is described below according to [3.112].

The utopia point F^* is the vector containing the individual global minima f_i^*, of the objectives and is defined as:

$$F^* = \{f_1^*, f_2^*, ..., f_k^*\}. \tag{3.5.3}$$

So, for the example above

$$F^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Let x_i^* be the respective global minimizers of $f_i(x), i = 1,...,k$, let $F_i^* = F(x_i^*), i = 1,...,k$ and let the 'pay-off' matrix, $\phi(i,i) = F_i^* - F^*, i = 1,...,k$. Then, the set of points in $\Re^k$, that are convex combinations of ϕ, i.e. $\phi\beta + F^*$, are referred to as the CHIM described earlier (where the addition of F^* is to shift the origin to F^*). So, for the example above

$$\phi = \left[\begin{bmatrix}1\\9\end{bmatrix} - \begin{bmatrix}1\\0\end{bmatrix} \quad \begin{bmatrix}10\\0\end{bmatrix} - \begin{bmatrix}1\\0\end{bmatrix}\right] = \begin{bmatrix}0 & 9\\9 & 0\end{bmatrix}$$

with $i = 17, \beta = (0.452, 0.548)$, the coordinates of point 17 on the CHIM is calculated as follows

$$\phi\beta + F^* = \begin{bmatrix}0 & 9\\9 & 0\end{bmatrix}\begin{bmatrix}0.452\\0.548\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}5.935\\4.065\end{bmatrix}.$$

Let $\underline{n}$ denote the unit normal to the CHIM pointing towards the origin from the point i, then $\phi\beta + t\underline{n},\ t \in \Re$ represents the set of points on that normal. Therefore, the sub-problem is to maximize t, whilst remaining on the normal. Meaning, that in order for a solution $F(x)$, to lie on the normal then $\phi\beta + t\underline{n} + F^* = F(x)$. Hence, the problem can be stated as follows:

$$\begin{aligned} &\text{Maximize} \quad t \\ &\text{Subject to:} \quad \phi\beta + t\underline{n} + F^* = F(x) \end{aligned} \tag{3.5.4}$$

The algorithm used in this work to implement the NBI method is as follows:

1. Calculate the current normal vector $\underline{n}$ to the CHIM pointing towards origin
2. Make t a vector $\underline{t}$, and it follows that

$$\underline{t} = \frac{F(x) - F^* - \phi\beta}{\underline{n}}.$$

3. If $t_1 = t_2$ then the current solution $F(x)$, lies normal to the CHIM.
4. Let $\underline{a} = F(x) - F^* - \phi\beta$, so that, $\underline{a}$ is the position vector of $F(x)$ from the current point on the CHIM.
5. Calculate the magnitudes of $\underline{a}$ and $\underline{n}$.
6. Calculate the angle between $\underline{a}$ and $\underline{n}$.
7. Calculate the distance along $\underline{n}$ of $F(x)$ and use this as the objective function to maximize.
8. Calculate the perpendicular distance of $F(x)$ from $\underline{n}$ and use this as the constraint. Since, if the perpendicular distance is zero, then the solution lies on the normal.

3.5.3 PHYSICAL PROGRAMMING METHOD

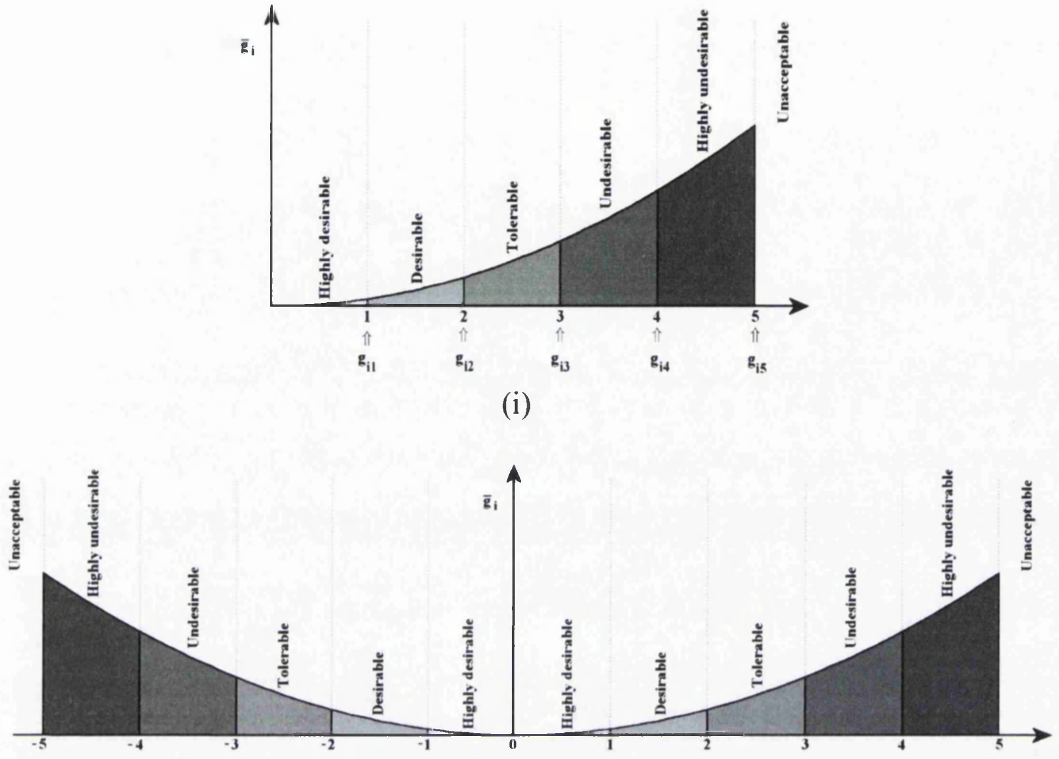
In general, to implement Physical Programming (PP), the solutions are mapped into another design space that captures the designer's preferences. The preference settings for the each of the multi-objectives are defined (ranging from highly desirable to highly undesirable) and then the performance (characterized by the original objective functions) is mapped into the new preference-based design space according to its 'desirability'. Optimization is then used to find the most 'highly desirable' solution.

PP has been applied to RDO in [3.114] and was first considered in [3.108]. The advantage of using PP over methods such as 'Weighted-Sums' to generate the Pareto set, is that the weight settings are not chosen using an iterative process. PP does not require weights to solve the problem. Instead the performance is mapped onto a non-dimensional, unimodal, 'class' function that captures the designer's performance preferences. Figure 3.5.6 shows two typical class functions with desirability regions for (i) the 'smaller the better' and (ii) the 'value-is-better' (for other cases such as 'larger-is-better' consult [3.108] for further details). The intervals vary from highly desirable → desirable → tolerable → undesirable → highly undesirable and anything greater than highly undesirable is considered unacceptable. The class function can take any form, as an example for the 'smaller-the better' scenario shown in Figure 3.5.6(i) it could take the following form:

$$\bar{g}_i = g_i^2 \qquad (3.5.5)$$

where $\bar{g}_i$ is the class function and g_i is the value of the ith objective function transformed according to the designer preferences.

From Figure 3.5.6 it can be seen that the 'value-is-better' desirability regions are the same as for 'smaller-the better' but with additional regions mirrored in the vertical axis. An example transformation for the 'smaller-the-better' scenario is described in Appendix A1. It should be noted, that if the same function is used for each range then it is sufficient to specify the ideal value, i.e. where $\bar{g}_i = 0$ and the unacceptable value, and the mapping of g_i onto the class function can be then done between those extremes. However, it may be desirable to use different functions in each region and so the preference regions need to be specified and, in such cases, convexity of the overall class function must be maintained.



(i)

(ii)

Figure 3.5.6: Class functions for (i) smaller-the-better; (ii) value-is-better. The horizontal axis is the ith objective function, g_i transformed according to designer preferences; the vertical axis is the class function, g_i to be minimized

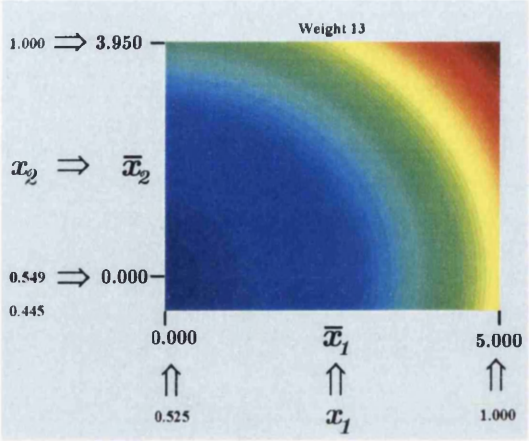
For the implementation here, the original design space is mapped into a surrogate design space (SDP), which represents the designer's preferences; the optimizer then uses this. Two possible SDPs are as follows

$$F(x) = \overline{x}_1^{\,2} + \overline{x}_2^{\,2}, \tag{3.5.6}$$

$$F(x) = (\overline{x}_1 - \overline{x}_2)^2 + \overline{x}_1 + \overline{x}_2\,. \tag{3.5.7}$$

It should be noted, that the link between the original design space and the SDP is based purely on the preferences and there is no physical link between them. For example, to map the designer's preferences into a SDP, for weight 13, say, the ideal values for x_1 and x_2, i.e. x_1^* and x_2^* could be $x_1 = 0.525$ and $x_2 = 0.549$. These preferences are mapped into a SDP, such as Equation 3.5.6, as is shown in Figure 3.5.7. The optimizer then only sees this SDP function and so the 'artificial' global minimum can be located in the desired position in the original design space. For the example shown in Figure

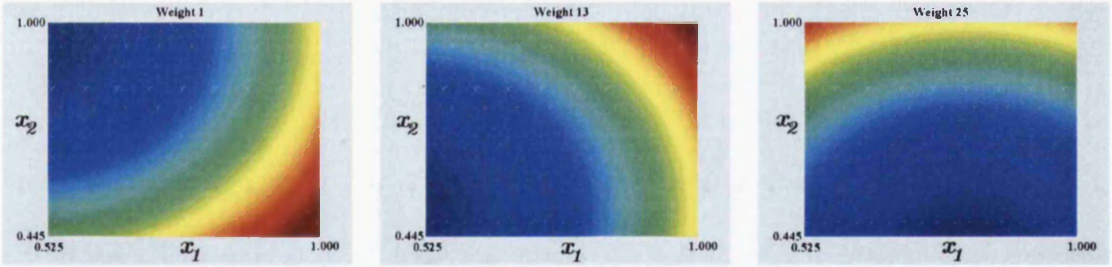
3.5.7 the optimizer may find a value of $\overline{x}_1 = 0.000$ and $\overline{x}_2 = 0.000$, corresponding to an actual value of $x_1 = 0.525$ and $x_2 = 0.549$, which are the ideal values for this case.



$x_1^* = 0.525$, $x_2^* = 0.549$

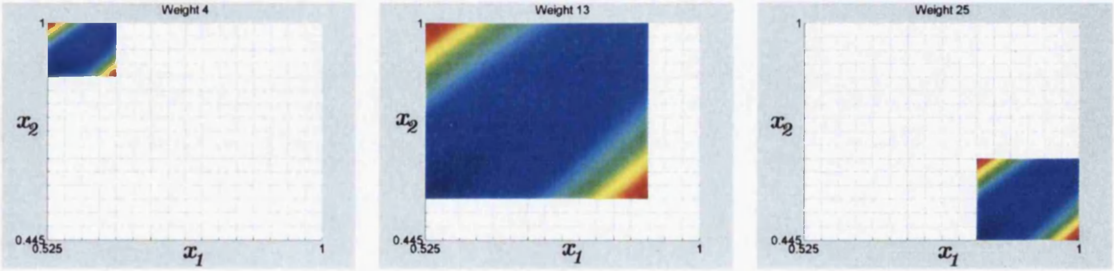
Figure 3.5.7: Example of mapping the designer's preferences onto the SDP given by Equation 3.5.6 (blue = low and red = high)

Figures 3.5.8 and 3.5.9 show various SDPs for different designer preferences or weightings that are transformed in a similar manner to the example in Figure 3.5.7, using Equations 3.5.6 and 3.5.7 respectively.



(a) $x_1^* = 0.525$, $x_2^* = 0.965$ (b) $x_1^* = 0.525$, $x_2^* = 0.549$ (c) $x_1^* = 0.703$, $x_2^* = 0.445$

Figure 3.5.8: Resulting SDP contour plots of Equation 3.5.6 for (a) weight 1; (b) weight 13; (c) weight 25; (blue = low and red = high)



(a) $x_1^* = 0.525$, $x_2^* = 0.861$ (b) $x_1^* = 0.525$, $x_2^* = 0.549$ (c) $x_1^* = 0.703$, $x_2^* = 0.445$

Figure 3.5.9. Resulting SDP contour plots of Equation 3.5.7 (a) weight 4; (b) weight 13; (c) weight 25; (blue = low and red = high)

3.5.4 Pareto front marching method

The use of the NBI and the PP methods described above to generate a Pareto set of solutions requires n separate optimization runs to determine each of the n Pareto points. Typically, when using a gradient-based optimizer the choice of a good starting point is required in order to locate the global rather than a local optimum solution. Choosing starting points for each of the separate runs is cumbersome especially as a good starting point to locate one Pareto point may not be a good starting point for locating another. In other words, starting at the same point for each optimization run does not guarantee that the optimizer will locate the global optimum for each point, instead it may find a local minimum. Furthermore, computational effort is wasted in searching the design space from the same point each time. In order to overcome these shortfalls, the Pareto front marching method (PFM-PP) was introduced in this work. It uses the utopia points as the starting points and aims to 'march' along the Pareto front by using the previous Pareto solution as the starting point for calculating the next Pareto point. In this way, the starting point is chosen in a systematic manner and is located in the vicinity of the next Pareto solution, therefore computational effort is not wasted in moving from some arbitrary starting point to the vicinity of the optimum.

PFM-PP was developed in Chapter 5 where the design space is transformed in the same manner as for PP (above) using SDPs similar to those in Figures 3.5.8 and 3.5.9. It can be seen from the results in Chapter 5, that PFM-PP generates a uniform distribution of points resulting in a smooth Pareto front. It also requires five-times fewer iterations to generate the solutions than the current NBI and PP implementations. PFM-PP can therefore be considered an efficient tool for generating robust designs.

3.5.5 Two-bar truss problem

A bi-objective formulation [3.106] of the two-truss problem of Kirsch [3.105] is used to test the implementations of the MO methods described in Sections 3.5.2 to 3.5.4. The problem is shown in Figure 3.5.10 and consists of determining the vertical position of point C and the cross-sectional areas of links AC and BC to minimize the total volume of the structure and to minimize the tensile stress in link AC. The design variables, which are all continuous, are x_1, x_2 and y. The constraints are the maximum allowable stresses in AC and BC that should not exceed 100,000 kPa and the total volume of the material used in the structure that should be held less than 0.1m^3. The problem formulation is shown below

$$\begin{aligned} \text{minimize:} \quad & f_{volume} = x_1\sqrt{(16+y^2)} + x_2\sqrt{(1+y^2)} \qquad (3.5.8) \\ \text{minimize:} \quad & f_{stress,AC} = \frac{20\sqrt{(16+y^2)}}{yx_1} \\ \text{subject to:} \quad & f_{volume} \le 0.1 \\ & f_{stress,AC} \le 100000 \\ & f_{stress,BC} \le 100000 \\ & 1 \le y \le 3 \\ & x_1, x_2 > 0 \\ \text{where} \quad & f_{stress,BC} = \frac{80\sqrt{(1+y^2)}}{yx_2} \end{aligned}$$

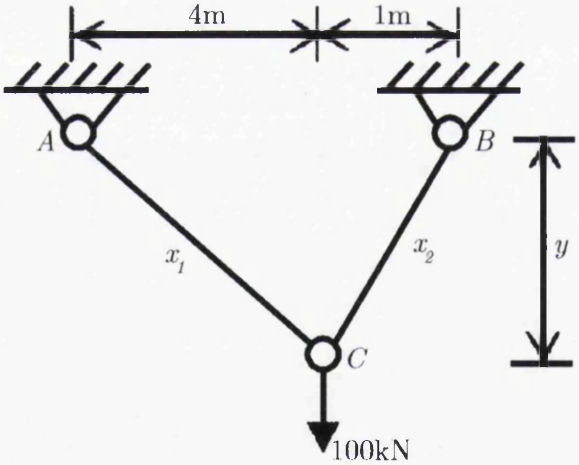


Figure 3.5.10: Two-bar truss [3.106]

The results to the two-bar truss problem, using the current implementations of NBI, PP and PFM-PP, are shown together with the result gained used E-MOGA in Ref. [3.106] in Figure 3.5.11. It can be seen, that the methods generate uniformly distributed Pareto points. NBI and PP require seven-times and 1.6-times more function evaluations respectively, than PFM-PP to generate the solutions. However, PFM-PP requires 1.8-

times more function evaluations than E-MOGA. It can be concluded that the current NBI, PP and PFM-PP implementations produce results that are consistent with previously published results and are suitable for application to other MO problems.

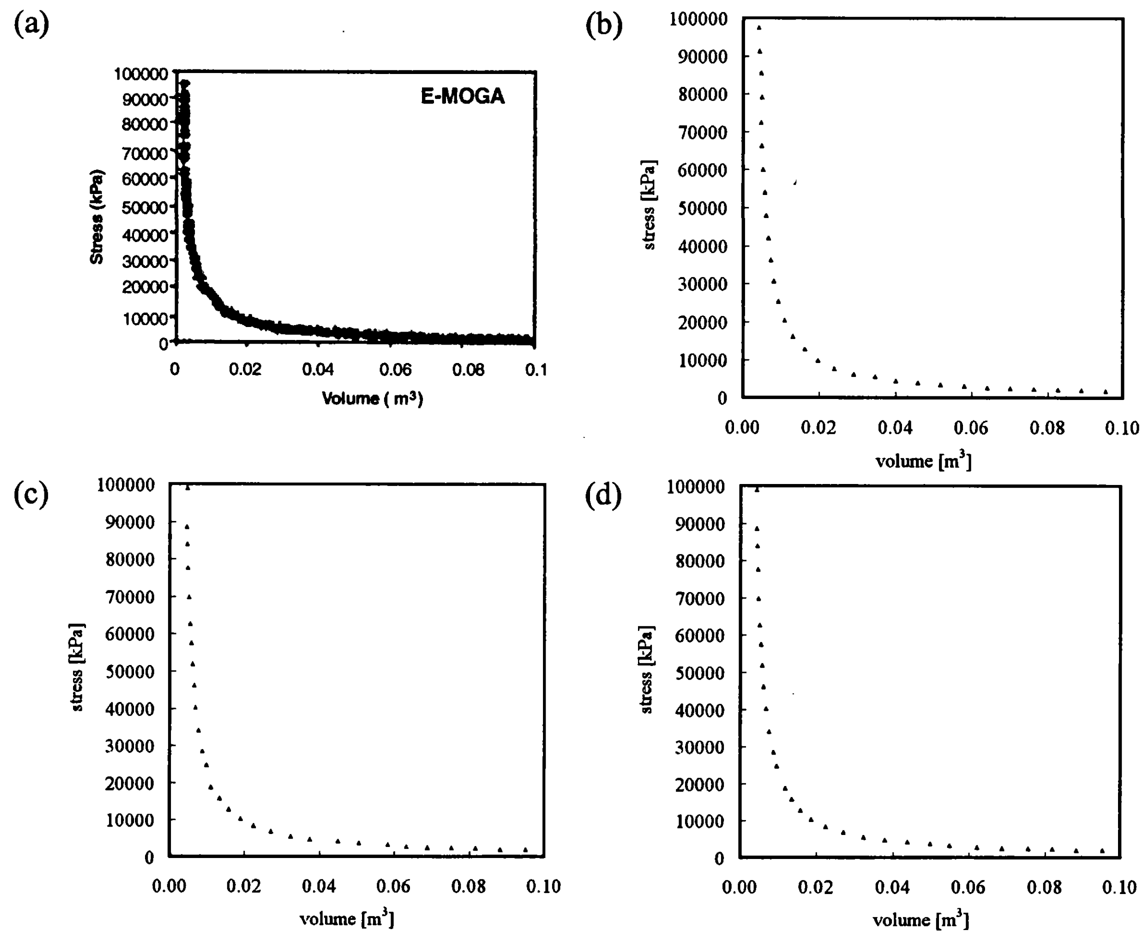


Figure 3.5.11: Pareto solution sets for the two-bar truss example: (a) E-MOGA result; 550 function calls [3.106], (b) NBI result; 7104 function calls, (c) PP result; 1606 function calls, (d) PFM-PP result; 980 function calls

3.6 CONCLUDING REMARKS

This chapter has given an overview of optimization modelling and some current applications of the various techniques. A number of the techniques described have been used elsewhere in this thesis. Size optimization, mid-range approximations, design of experiments, and multi-objective optimization methods have been used in Chapter 5; in Chapter 6, the SIMP topology optimization method, shape optimization, design space approximation, manufacturing constraints, and minimum member size control have been applied.

Two new pieces of research are described in this Chapter in Sections 3.4.2 and 3.5.4:

1) During the research carried out in Chapter 5, the OLH technique was used in a robust design optimization loop (Chapter 4) and it was deemed necessary to develop a technique to formulate it. A new technique has been developed to formulate this DoE using a permutation GA; it is described in detail in Section 3.4.2. It has been shown, that the formulation of OLHs is ideally suited to using GAs since the problem uses discrete design variables; other optimization methods such as integer programming methods have not been considered. It can be seen, that a binary encoding, and the inherent need for a penalization method, can be used to solve the problems. But, it has also been shown that the use of a permutation encoding is better suited to the problem, because it does not 'waste' computational effort gradually filtering out infeasible solutions from the population; the permutation encoding requires fewer function evaluations and produces improved solutions. Further work is necessary to fully assess the quality of the results found using the permGA and the Audze-Eglais objective function. Using other objective functions, such as those discussed in Section 3.4.2.1 can achieve this. Overall, it can be concluded, that the permGA method is an effective tool for developing OLH DoE.

2) Also, during the work of Chapter 5, a new multi-objective method was developed. This method is called the Pareto front marching method and is based on the physical programming method; it is described in Section 3.5.4. It can be considered as an efficient tool for MO optimization.

3.7 References

[3.1] Bendsøe M.P., Sigmund O., "Topology optimization: theory, methods and applications", Springer, ISBN: 3-540-42992-1, 2003.

[3.2] Hassani B., Hinton E., "Homogenization and structural topology optimization. Theory, practice and software", pp. 4, Spring, ISBN: 3-540-76211-6, 1999.

[3.3] Hinton E., Sienz J. and Hassani B., "Fully integrated design optimisation for engineering structures", 3rd Int. Conf. on Comp. Struct. and Tech., Budapest, Hungary, 21-23, Edinburgh, UK, 1996.

[3.4] Hemp W.S. "Optimum Structures", Clarendon Press, Oxford, 1973.

[3.5] Hinton E., Sienz J., Özakça M., 'Analysis and Optimization of Prismatic and Axisymmetric Shell Structures: Theory, Practice and Software', Springer Verlag, ISBN: 1852334215, 2003.

[3.6] Rozvany G. "review article: Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics" Struct Multidisc Optim, Volume 21, Issue 2, pp. 90-108, 2001.

[3.7] Rozvany G.I.N., "Stress ratio and compliance based methods in topology optimization –a critical review"Struct Multidisc.Optim.21 ,109 –119, 2001.

[3.8] Mattheck C., Burkhardt D., Erl D., "Shape optimization of engineering components by adaptive biological growth", Conf. Engineering optimization in design processes, pp.15–26, Springer, 1990.

[3.9] Xie Y.M., Steven G.P., "Evolutionary structural optimization", Springer Verlag, Berlin, 1997.

[3.10] Querin O.M., Lencus A., "Optimisation of Structures Subject to Multiple Load Cases Using Biological Growth Laws", Proc. Australasian Conf. on Structural Optimization, pp. 241-250, University of Sydney, Australia, February 11-13, 1998.

[3.11] Li Q., "Evolutionary Structural Optimization for Thermal and Mechanical Problems", PhD Thesis, University of Sydney, Australia, March 2000.

[3.12] Zhao C., Hornby P., Steven G.P., Xie Y.M., "A Generalized Evolutionary Method for Natural Frequency Optimization of Membrane Vibration Problems in Finite Element Analysis", Computers and Structures, 66, pp. 353-364, 1998.

[3.13] Querin O.M., Steven G.P., Xie Y.M., "Evolutionary structural optimization (ESO) using a bi-directional algorithm", Eng. Comp., 15 ,1031-1048, 1998.

[3.14] Querin O.M., Steven G.P., Xie Y.M., "Morphing Evolutionary Structural Optimization (MESO)", in Engineering Design Optimization – Proc. 1st ASMO UK / ISSMO conf., 8-9 July, Ilkley, UK, MCB University press, pp.311-316, 1999.

[3.15] Lencus A., Querin O.M., Steven G.P., Xie Y.M., "Modifications to the Evolutionary Structural Optimisation (ESO) Method to Support Configurational Optimisation", 06-EVM1-5, proc. 3rd WCSMO, Buffalo, NY, 1999.

[3.16] Kim H., Querin O.M., Steven G.P., Xie Y.M., "Development of an Intelligent Cavity Creation (ICC) Algorithm For Evolutionary Structural Optimisation", Proc. Australasian Conf. on Structural Optimization, pp. 241-250, University of Sydney, Australia, February 11-13, 1998.

[3.17] Cervera E., Trevelyan J., "An Alternative Approach to Evolutionary Structural Optimisation (ESO) using the Boundary Element Method and B-Spline Representation", Proceedings of the 4th ASMO UK / ISSMO Conference, Newcastle, UK, 4th-5th July, pp.249-256, 2002.

[3.18] Zheng Q.Z., Querin O.M., Barton D.C., Brodie R.N., "Topology Optimization using the Genetic Programming Method", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.227-228, 2003.

[3.19] Woon S.Y., Tong L., Querin O.M., Steven G.P., "Optimizing Topologies through a Multi-GA System", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.229-230, 2003.

[3.20] Bendsøe M.P., Kikuchi N., "Generating optimal topologies in structural design using a homogenization method.", Comp. Meth. Appl. Mech. Eng., 71, 197-224, 1988.

[3.21] Chiandussi G., Gaviglio I., Ibba A., "Topology Automotive Component Design by Optimality Criteria and given Lagrange Multiplier", in Proc. of the 3rd Int. Conf. on Engineering Computational Technology, Stirling, United Kingdom, paper 67, 2002.

[3.22] Fleury C., "CONLIN: An efficient dual optimizer based on convex approximation concepts", Structural Optimization, 1, pp. 81-89, 1989.

[3.23] Svanberg K., "The method of moving asymptotes – a new method for structural optimization", Int. J.Numerical Methods in Engineering, 24, pp. 359-373, 1987.

[3.24] Bendsøe M.P., "Optimal shape design as a material distribution problem", Struct. Optim., 1 , pp193–202,1989.

[3.25] Sigmund O., Petersson J., "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima", Structural Optimization, 16, 68-75, 1998.

[3.26] Zhou M., Shyy Y. K., Thomas H. L., "Checkerboard and Minimum Member Size Control in Topology Optimization", 29-TOM2-6, WCSMO 3, Buffalo, NY, 1999.

[3.27] Ambrosio L., Buttazzo G., "An optimal design problem with perimeter penalization", Calc. Var., 1, pp55-69, 1993.

[3.28] Haber R. B., Bendsøe M. P. Jog C., "A new approach to variable-topology shape design using a constraint on the perimeter", Structural Optimization, 11, pp.1-12, 1996.

[3.29] Sigmund O., "Design of material structures using topology optimization", PhD Thesis, Dept. of Solid Mechanics, Technical University of Denmark, 1994.

[3.30] Petersson J., Sigmund O.,"Slope Constrained Topology Optimization", Int. J. for Numerical Methods in Engineering, 41, pp 1417-1434, 1998.

[3.31] "Altair OptiStruct Version 6.0', Altair Engineering Limited, 2003.

[3.32] Personal communication with Zhou M., Altair Engineering Inc., 2003.

[3.33] Zhou M., Shyy Y.K. and Thomas H.L. "Topology Optimization with Manufacturing Constraints", Proc. 4th WCSMO conf., Dalian, China, June 2001.

[3.34] Dornberger, R., Büche., Stoll, P. "Multidisciplinary optmization in turbomachinery design." ECCOMAS, 2000.

[3.35] Golovidov et al "A flexible, object-based implementation of approximation models in an MDO (multi-disciplinary optimization) framework". Design Optimization: International journal for Product & Process improvement, Vol.1 No. 4, 1999.

[3.36] Barthelemy J.-F.M. and Haftka R.T., "Approximation concepts for optimum structural design - a review", Structural Optimization, 5, 129-144, 1993.

[3.37] Toropov, V.V.; Alvarez, L.F. Approximation model building for design optimization using genetic programming methodology. Paper AIAA-98-4769, 7th AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization, St. Louis (USA), , part 1, pp. 490-498, AIAA, 1998. ISBN 1-56347-273-2, September 2-4, 1998. http://www.brad.ac.uk/staff/vtoropov/burgeon/b_mars.htm

[3.38] Sobieszczanski-Sobieski J., Haftka R.T., "Multidisciplinary aerospace design optimization: survey of recent developments". Struct.Optim. 14 , pp. 1–23, 1997.

[3.39] Jin R., Chen W., Simpson T.W., "Comparative studies of metamodelling techniques under multiple modelling criteria" Struct Multidisc Optim 23 (2001) 1, 1-13.

[3.40] Giunta, A. A. " Aircraft Multidisciplinary Design Optimization Using Design of Experiments Theory and Response Surface Modeling Methods," Ph.D. Thesis, Virginia Tech., May 1997.

[3.41] Simpson, T., Peplinski, J.D., Koch, P.N., Allen, J.K, " On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments," ASME Design Theory and Methodology '97, ASME97-DETC97/DTM-3881, 1997.

[3.42] Simpson T. W., Booker A. J., Ghosh D., Giunta A. A., Koch P. N., Yang R.-J. "Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion," Struct. Multidisc. Optimization, accepted for publication 2003.

[3.43] Markine V.L., "Optimization of the dynamic behaviour of mechanical systems", PhD Thesis, Delft University of technology, 1999.

[3.44] Alvarez L.F., "Design optimization based on genetic programming: Approximation model building for design optimization using the response surface methodology and genetic programming", PhD Thesis, Uni. of Bradford, UK, 2000.

[3.45] Haftka R. T. “Combining global and local approximations”, AIAA Journal 29: 1523–1525,1991.

[3.46] Myers R.H., Montgomery D.C. “Response surface methodology: Process and Product Optimization Using Designed Experiments”, John Wiley & Sons, New York, NY, 1976.

[3.47] Box G.E.P., Draper N.R., “Empirical model-building and response surfaces”, New York: John Wiley and Sons, 1987.

[3.48] Koza J.R., “Genetic programming: on the programming of computers by means of natural selection”, Cambridge, Mass: MIT Press, ISBN: 0-2621-11705, 1992.

[3.49] Smith M. “Neural networks for statistical modelling”, NewYork: Von Nostrand Reinhold, 1993.

[3.50] Cheng B., Titterington D.M., ”Neural networks: a review from a statistical perspective”, Statistical Sci. 9 ,pp2–54, 1994.

[3.51] Sacks J., Welch W.J., Mitchell T.J., Wynn H.P., “Design and analysis of computer experiments”. Statistical Science 4, pp.409-435, 1989.

[3.52] Booker A.J., Dennis J.E., Frank P.D., Serafini D.B., Torczon V., Trosset M.W., “A rigorous framework for optimization of expensive functions by surrogates”, Struct.Optim. 17 , 1–13, 1999.

[3.53] Giunta, Anthony A., "Aircraft Multidisciplinary Design Optimization using Design of Experiments Theory and Response Surface Modelling Methods," MAD 97-05-01, May 1996.

[3.54] Toropov V.V., Alvarez L.F., “Application of genetic programming to the choice of structure of multipoint approximations”
www.brad.ac.uk/staff/vtoropov/conf/webconf.htm

[3.55] Lauridsen S., Madsen J.I., “Comparison of Global Approximation Strategies for Fluid Design Optimization” 24-AAM1-4, proc. 3rd WCSMO, Buffalo, NY, 1997.

[3.56] Carpenter W.C., Barthelemy J.-F.M., “A comparison of polynomial approximations and artificial neural nets as response surfaces”, Struc.Opt.,5, 1993.

[3.57] Sellar R. S., Batill S. M., and Renaud J. E., “Concurrent Subspace Optimization Using Gradient-Based Neural Network Response Surface Mappings”, AIAA Paper 96-4019, 6th AIAA/NASA/USAF/ISSMO Symposiumon Multidisciplinary Analysis and Optimization, Bellevue, Washington, September 1996.

[3.58] Liu W., Batill S.M. and Renaud, J.E., "Implementation Issues in Gradient-enhanced Neural Network Response Surface Approximations," Proc. 4th WCSMO conf., Dalian, China, June 2001.

[3.59] Liu, W. and Batill, S.M., "Gradient-Enhanced Neural Network Response Surface Approximations," AIAA Paper 2000-4923, 8th AIAA/NASA/USAF/ISSMO Multidisciplinary Analysis and Optimization, Long Beach, California, September 2000.

[3.60] Smith M., "Neural networks for statistical modelling", New York: Von Nostrand Reinhold, 1993.

[3.61] Giunta A., Watson L.T., "A comparison of approximation modelling techniques: Polynomial versus interpolating models", 7th AIAA/USAF/NASA/ISSMO Symp. Multidisciplinary Analysis & Optimization, vol. 1, pp 392–404, St. Louis, MO, 1998.

[3.62] Simpson, T. W., Mauery, T. M., Korte, J. J. and Mistree, F. "Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization," 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, St. Louis, MO, September 2-4, AIAA, 1 (381-391) AIAA-98-4755.

[3.63] Simpson T. W., Mauery T. M., Korte J. J., Mistree F., "Kriging Metamodels for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," AIAA Journal, 39:12 (2233-2241), 2001.

[3.64] Simpson T. W., Lin D.K.J., Chen W. "Sampling Strategies for Computer Experiments: Design and Analysis," International Journal of Reliability and Applications, 2:3 (209-240), 2001.

[3.65] Martin, J. D. and Simpson, T. W. (2003) A Study on the Use of Kriging Models to Approximate Deterministic Computer Models, ASME Design Engineering Technical Conferences - Design Automation Conference, Shimada, K., ed., Chicago, IL, September 2-6, ASME, Paper No. DETC2003/DAC-48762.

[3.66] Leary S.J., Bhaskar A., and Keane A.J., "A Knowledge-Based Approach to Response Surface Modelling in Multifidelity Optimization", J. of Global Optimization, 26, 297-319, 2003.

[3.67] Srivastava A., Hacker K., Lewis K.E., "Investigation of Different Approximation Techniques in the Design of the High Speed Civil Transport Aircraft", 24-AAM1-2, proc. 3rd WCSMO, Buffalo, NY, 1999.

[3.68] Toropov V.V., "Simulation approach to structural optimization". Structural Optimization, 1, No 1, pp. 37-46, 1989.

[3.69] Toropov V.V., Filatov A.A., Polynkin A.A., "Multiparameter structural optimization using FEM and multipoint explicit approximations", Structural Optimization, 6, 7-14,1993

[3.70] Toropov V.V., van Keulen F., Markine V.L., de Boer H., "Refinements in the multi-point approximation method to reduce the effects of noisy responses", 6th AIAA/USAF/NASA/ISSMO Multidisc. Anal. Opt., Bellevue, Part 2, pp. 941-951,1996.

[3.71] Van Keulen F., Toropov V.V., "New developments in structural optimization using adaptive mesh refinement and multi-point approximations", Engineering Optimization, v. 29, 217-234, 1997.

[3.72] Sienz J., Bates S.J., Langley D.S., Pittman J.F.T. "Comparison of globally converging optimization with mid-range approximation based optimization applied to slit die design" Proc. 3rd ASMO UK / ISSMO conf., Harrogate, UK, 2001.

[3.73] Schoofs A.J.G., van Houten M.H., Etman L.F.P. van Campen D.H., "Global and mid-range function approximation for engineering optimization", ZOR/Mathematical Methods of Operations Research, Vol. 46, No. 3, pp. 335-359, 1997.

[3.74] Xu S. and Grandhi V., "Multipoint approximation for reducing the response surface model development cost in optimization", In: Toropov, V.V. (ed.), Proceedings of First ASMO UK / ISSMO Conference on Engineering Design Optimization, Ilkley, UK, pp. 381-388, 1999.

[3.75] Magazinović G., "Two-point Mid-range Approximation Enhanced Recursive Quadratic Programming Method", WCSMO 5, Lido di Jesolo, Italy, May 19-23, 2003.

[3.76] Venter, G. (1998), "Non-dimensional response surfaces for structural optimization with uncertainty", PhD thesis, University of Florida, USA.

[3.77] Fisher R.A., "Statistical Methods for Research Worker's, Oliver and Boyd, London, 1925.

[3.78] Kodiyalam,S., Lin, J.S., Wujek, B.A. "Design of experiments based response surface models for design optimization", Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Long Beach, CA, April 20-23, 1998.

[3.79] Mackay M. D., Beckman R. J., Conover W. J., "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code", Technometrics, Vol. 21, pp. 239-245, 1979.

[3.80] Fang K.-T., Lin D.K.J., Winker P., and Zhang Y., "Uniform design: Theory and application", Technometrics, 42:237–248, 2000.

[3.81] Kalagnanam J.R., Diwekar U.M., "An efficient sampling technique for off-line quality control". Technometrics, 39(3):308–319,1997.

[3.82] http://www.itl.nist.gov/div898/handbook/pri/pri.htm

[3.83] Box G.E.P., Wilson K.G., "On the experimental attainment of optimal conditions", Journal of the Royal Statistical Society, Series B, 13, 1-45, 1951.

[3.84] Box G.E.P., Behnken D.W., "Some new three level designs for the study of quantitative variables", Technometrics, Vol. 2, No. 4,1960, pp. 455-475.

[3.85] Iman R.L., Conover W.J., "Small sample sensitivity analysis techniques for computer models, with an application to risk assessment", Communications in Statistics, Part A. Theory and Methods, 17:1749-1842, 1980.

[3.86] Shewry M., Wynn H., "Maximum entropy design", J. Appl. Statist. 14 (2), pp. 165-170, 1987.

[3.87] Sacks J., Schiller S.B., Welch W.J., "Designs for computer experiments" Technometrics, Vol. 31 Issue 1, pp41-48, Feb 1989.

[3.88] Johnson M., Moore L., Ylvisaker D., "Minimax and maximin distance designs", J. Statist. Plann. Inference 26, pp. 131-148, 1990.

[3.89] Audze, P. Eglais, V. "New approach for planning out of experiments", Problems of Dynamics and Strengths, vol. 35 Riga, pp. 104-107 (Russian), 1977.

[3.90] Jin, R., Chen, W., and Sudjianto, A., "An Efficient Algorithm for Constructing Optimal Design of Computer Experiments" DETC-DAC48760, 2003 ASME Design Automation Conference, Chicago, IL, September 2-6, 2003.

[3.91] Goldberg D.E. "Genetic algorithms in Search, Optimization, and Machine Learning",Reading,Mass.:Addison-Wesley, 1989.

[3.92] Bates S. J., Sienz J. and. Langley D. S., "Formulation of the Audze–Eglais Uniform Latin Hypercube design of experiments", J. Advances in Engineering Software, vol. 34/8 pp. 493 – 506, 2003.

[3.93] Langley D.S., "Genetic algorithm focussed comparative optimization studies for a broad scope of engineering applications", PhD Thesis, Uni. of Wales Swansea, 2003.

[3.94] Michalewicz Z., "Genetic algorithms + data structures = evolution programs", Springer-Verlag, 1992.

[3.95] Bates S.J., Sienz J., Toropov V.V., "Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm", 45th AIAA Conference, Palm Springs, California, submitted 2003.

[3.96] Latin Hypercube sampling tool from web-page http://www.mathepi.com/epitools/lhs/nrpage.html , 2001.

[3.97] Personal communication from Prof. Toropov V.V., toropov@altair.com, 2003.

[3.98] Merriam-Webster online dictionary. http://www.m-w.com

[3.99] Oliver I.M., Smith D.J., Holland J.R.C., "A study of permutation crossover operators on the travelling salesman problem", Proc. 2nd Int. Conf. on Genetic Algorithms, Massachusetts Institute of Technology, Cambridge, MA, pp411-423, 1987.

[3.100] Goldberg D.E., Lingle R., "Alleles, Loci, and the TSP", proc. 1st Int. conf. on Genetic Algorithms, Hillsdale, NJ, pp154-159, 1985.

[3.101] Davis L., "Applying adaptive algorithms to epistatic domains", proc. Int. Joint conf. on Artificial Intelligence, pp162-164, 1985.

[3.102] Pareto V. "Manual of Political Economy", 1971 translation of 1927 edition, New York: Augustus M. Kelley, 1906.

[3.103] Messac A., Ismail-Yahaya A., and Mattson C. A., "The Normalized Normal ConstraintMethod for Generating the Pareto Frontier,la Structural Optimization, Vol. 25, No. 2, pp. 86-98, 2003.

[3.104] Chen. W., Wiecek, M., and Zhang, J., "Quality Utility: A Compromise Programming Approach to Robust Design", accepted by ASME Journal of Mechanical Design, (September, 1998).

[3.105] Kirsch U., "Optimal structural design", New York, NY: McGraw-Hill, 1981.

[3.106] Farhang-Mehr A., Azarm S., "Entropy-based multi-objective genetic algorithm for design optimization", Struct Multidisc Optim 24, 351–361 Springer-Verlag 2002.

[3.107] Das, I. and Dennis, J.E., 1997, "A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto set Generation in Multicriteria Optimization Problems," Structural Optimization, Vol. 14, pp. 63-69.

[3.108] Messac, A., "Physical Programming: Effective Optimization for Computational Design," AIAA Journal, Vol. 34, No. 1, Jan. 1996, pp. 149-158.

[3.109] Ismail-Yahaya A., Messac A., "Effective generation of the Pareto frontier using the normal constraint method", 40th Aerospace Sciences Meeting and Exhibit Paper No.AIAA2002-0178, Reno, Nevada, 2002.

[3.110] Fonseca C. M., Fleming P.J. "Genetic Algoritms for multiobjective optimisation: formulation , discussion, and generalisation". Proceedings of the fifth International Conference on Genetic Algorithms. 416-423, 1993.

[3.111] Narayanan S., Azarm S., “On improving multi-objective genetic algorithms for design optimization”, Struct. Optim., 18, pp146–155, 1999.

[3.112] Das I., Dennis J., "Normal-Boundary Intersection: A new method for generating Pareto optimal points in multicriteria optimization problems", SIAM J. on Optimization Vol 8 No. 3 pp. 631-657, August 1998.

[3.113] Das, I. “An Improved Technique for Choosing Parameters for Pareto Surface Generation using Normal-Boundary Intersection” 28-MOM2-2, WCSMO 3, Buffalo 1999.

[3.114] Messac, A., and Sundararaj, G. J., “A Robust Design Approach using Physical Programming”, 38th Aerospace Science Meeting, Paper No. AIAA 2000-0562, Reno Nevada, January 10-13, 2000.

CHAPTER 4

ROBUST DESIGN OPTIMIZATION AND OTHER 'DESIGN FOR UNCERTAINTY' TECHNIQUES

Summary: This chapter gives an overview of the robust design methodology developed by Taguchi who described it as *parameter design*. He introduced the notion of accounting for scatter caused by uncontrollable system parameters through the identification of 'noise' factors. A brief description of the method and a review of some of the literature regarding the limitations of Taguchi's approach are given. The original method was conceived for use in physical tests. Nowadays computational experimentation is used and a discussion of several computer oriented robust design approaches is given. Following on, a description of the computer implementation of the robust design procedure used in the slit die problem in Chapter 5 is given. A discussion on other methods for designing under uncertainty such as reliability-based optimization, Monte Carlo simulation, and six-sigma is presented for completeness.

4.1 Introduction

Section 4.1: Gives an introduction to the chapter by discussing

(i) the progression from ideal deterministic optimization to robust design optimization, which accounts for uncertainties in a system's parameters.

(ii) the progression from physical off-line simulation to virtual computer-based simulation and optimization.

Section 4.2: Discusses noise and control factors and the modelling of uncertain variations. A discussion on how uniform and normally distributed sample points are implemented in this work is given.

Section 4.3: Gives a brief summary of the Taguchi robust design methodology and its limitations. Discussions on some current methods in robust design are also given with concentration on the Robust concept exploration method, which is an advancement of the original Taguchi method.

Section 4.4: Describes the implementation of the robust design procedure that has been used in this work (Chapter 5).

Section 4.5: Discusses other methods for stochastic optimization including the methods under the category of 'computational stochastic mechanics' including reliability-based optimization, perturbation techniques and Monte Carlo simulations. Finally a discussion on the six-sigma management process is given.

Section 4.6: Gives some concluding remarks.

Section 4.7: Lists the references used in the chapter.

So far, in this Thesis only deterministic optimization has been considered. This is principally due to insufficient computational power being available to apply the methods to account for realistic variation in the system parameters. "In the 1990s, research in design optimization via computer simulation tended to ignore uncertainty about physical parameters, manufacturing conditions, operating conditions, etc.; today, engineers are seriously seeking methods to manage uncertainty" [4.1]. So engineering design – especially structural optimization-based design – is moving beyond formulating the problem assuming deterministic parameters only.

In conventional deterministic optimization the objective is to minimize or maximize a function subject to constraints. The optimum is calculated under ideal conditions that are difficult to achieve in reality and real systems always exhibit uncertain deviations from the nominal state, termed here as scatter. Scatter is caused by uncertain variations in the design variables (control factors) and/or the noise variables (factors beyond the control of the designer or expensive to control) of the real system. To date designers have been reluctant to accept and implement designs found using deterministic optimization because they are often sensitive to the scatter present in real systems, they trust test results more than analysis results because decisions are made for the physical and not for the virtual world. This is illogical because physical and computational results will not be exactly the same: "even a physical test repeated several times under the same conditions will never produce the same result either. This is caused by the unavoidable scatter of the properties of the system examined and the boundary conditions of the test" [4.2] and also variation in the testing processes. Scatter arises from numerous sources such as the operating conditions, the loading conditions, the raw material and the manufacture. "The optimum obtained in the conventional formulation of deterministic structural optimization problems may become practically not applicable or unreliable due to the deviation of the actual structural response from the computational results obtained with the nominal values of the design variables and parameters. Thus it is reasonable to account for the randomness of the structural parameters in the optimum design" [4.3]. The methodologies concerned with accounting for scatter in a virtual design optimization environment can be broadly categorized under the heading stochastic design optimization.

Figure 4.1 illustrates the difference between a deterministic solution and a solution that accounts for variation in the system parameter, x, i.e. a more robust solution. The target in this case is to minimize the objective function. It can be seen, that the deterministic optimum solution is the value of x corresponding to the minimum value of the objective function (or performance characteristic), $f(x)$. If there is some variation of the system parameter, Δx, then it can be seen, that the resultant variation in the objective function, $\Delta f(x)$, causes the performance to become intolerable. A more 'robust' solution is indicated and it can be seen, that the variation in performance of this solution is less than that of the deterministic solution and it is less sensitive to the same Δx.

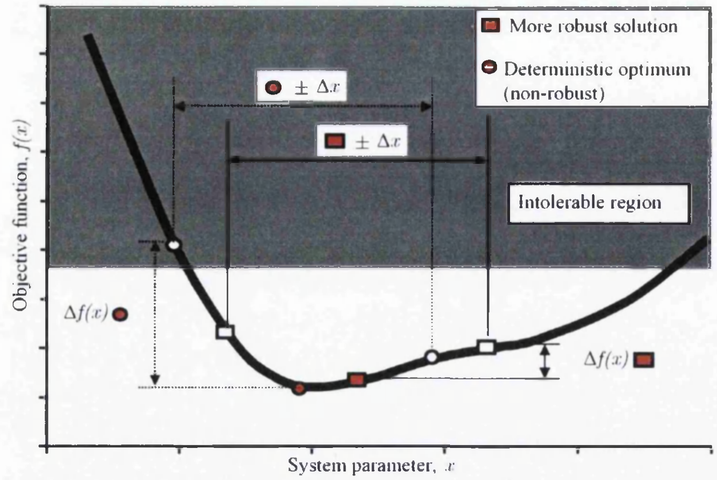


Figure 4.1: Comparison of a deterministic, non-robust optimum to a more robust solution

The main aim of an engineer is to develop durable and reliable products. Robust design optimization (RDO) is one area that is receiving considerable interest and this chapter is primarily concerned with RDO. RDO is one way of addressing the issue of uncertainty in the designer's mind by giving confidence in the simulation-based results; it takes into account uncertainty in a design response characteristic due to the noise factors (other methods are discussed in Section 4.5).

Taguchi [4.4] conceived the robust design idea with the aim to improve 'quality' by minimizing a product's sensitivity to variation without eliminating the causes of the variation. The original method is described in Section 4.2; it is essentially a method for designing processes or products that have consistent, high-level performance despite being subjected to a wide range of changing customer and manufacturing conditions, i.e. products and processes that are minimally sensitive to uncontrollable factors.

Since the aim of RDO is to minimize or maximize the primary objective function and minimize the sensitivity of the solution to scatter, the problem is bi-objective. This requires, that not only the performance of the solution be brought towards a target but also that the variation from the target is minimized. "It is generally recognized that the robustness of a design objective can be achieved by simultaneously 'optimizing the mean performance' and 'minimizing the performance variance'."[4.5] Minimizing both objectives simultaneously is not usually possible and a compromise between improvements in performance variation and in mean performance is required.

There are two ways to minimize variability, the first is to eliminate the source of the noise – this can be very costly and time consuming – the other is to minimize the product's sensitivity to noise. The latter approach is adopted in robust design leading to the following definition of robust design:

"A product or process is said to be robust when it is insensitive to the effects of sources of variability, even though the sources themselves have not been eliminated." [4.6]

Taguchi's robust design method was developed for use with experimentations carried out 'off-line' in a physical environment where, to some extent, the system parameters can be controlled, although uncertain variations in the response will still remain. The experimentation techniques are limited to testing parameters at two or three levels (typically). "If random error of important size is present, then any functional dependence of the error variance on the control factors would have to be modelled" Sacks and Welch [4.7]. In a virtual environment the simulations are deterministic since the system parameters can be adjusted to a desired level and the response has no variation if the calculation is repeated with the same boundary conditions. As such, "complex relationships can be uncovered with far fewer observations when random error is absent" Sacks and Welch [4.7]. Although the Taguchi method can be applied to computer experiments, the capability to simulate the response deterministically for any set of system parameters will not be fully exploited using the original approach.

Reuter *et al.* [4.2] observe, that "normally the scatter of the outputs is expected to have about the same order of magnitude as the scatter of the inputs. If it is considerably higher, it means that an amplification of scatter has occurred. In many cases the amplification originates from the (computer) simulation model itself and is due to numerical problems in the respective model and/or solver. This can be recognized by the strongly scattered results that do not correlate to physical parameters. The simulation model has to be modified in a case like this, as otherwise the results depend on chance alone and are therefore irrelevant. In cases that show an obvious correlation, a vast scatter hints at an unfavourable design of the system, as it reacts to small scatter in the system parameters with a large scatter. A system like this is hard to control in practice: it can only be controlled be keeping the scatter of the system parameters low. But this means nothing else but the use of very tight tolerances, a thing that is expensive and labour-intensive, if not impossible."

Welch and Sacks [4.8, 4.7] give a summary of work on quality improvement via computer experiments. The argument they provide is similar to that discussed in Chapter 3 on 'design and analysis of computer experiments' (DACE).

This chapter focuses on robust design using computer simulations; some other computational methods for designing under uncertain conditions are discussed in Section 4.5 together with a currently popular management process called six-sigma.

4.2 NOISE AND CONTROL FACTORS

Control factors are the design variables of a parameterized problem. In deterministic optimization, these are the only factors that influence the optimization process. In a stochastic optimization process there are additional un-controllable factors called *noise factors* and the variation of these factors is responsible for scatter in the performance of a product (discussed in Section 4.1).

Noise factors are the uncontrollable system parameters, e.g. in the case of the slit die design problem presented in Chapter 5 the noise factors are uncontrollable (or too expensive to completely control) variations in the operating conditions, such as the operating temperature and the flow rate of polymer into the slit die. Other factors include the variation of the raw material and in the manufacturing of a component. In the case of the A-pillar studied in Chapter 6 some possible uncontrollable factors could be the angle of impact of the test plate on the A-pillar or the stress-strain properties of the material. In any case, the sources of system noise need to be identified.

4.2.1 DISTRIBUTIONS OF FACTORS

In a computational simulation it is possible to fully control all of the system parameters, i.e. the settings of the controllable and uncontrollable factors. So any distribution of the uncertain variation of the system parameters can be modelled. For example, it may be observed from physical testing that the variation of operating temperature in a manufacturing process is normally distributed and the variation between ± three standard deviations is ± 10°C from the nominal temperature. This distribution can be applied to the computer model.

In order to determine the impact of the variations of the system parameters on the performance, a sample is taken using for example, Monte Carlo simulation or a Latin Hypercube (Chapter 3) sampling technique. Two examples are presented below for generating a sample, the first assumes that the variation of a factor is uniformly distributed between two limits, and for the second it is assumed that it is normally distributed. These two distributions are used in the RDO problem in Chapter 5. Other distributions include triangular, Weibull, log-normal, asymmetric or some known random distribution.

UNIFORM DISTRIBUTION

Figure 4.2 shows a uniform distribution for variable x_i between the bounds x_i^l and x_i^u. Using this distribution x_i is equally likely to be any value between and including these bounds.

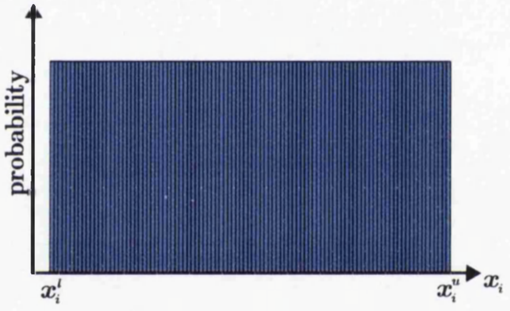


Figure 4.2: Uniform distribution of points

For example, the uniform distribution of points for a sample size of 99 is as shown in Figure 4.3.

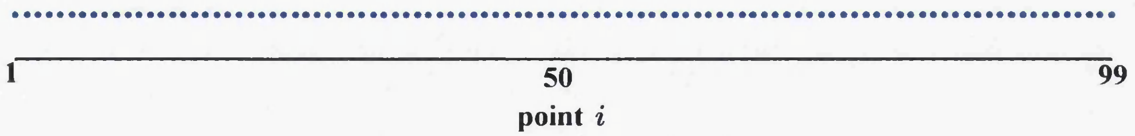


Figure 4.3: 99 uniformly distributed points

NORMAL DISTRIBUTION

In order to generate a normal distribution of points, the area underneath the normal distribution curve is divided into n regions of equal area, i.e. equal probability regions (the cumulative distribution function). For example, Figure 4.4 shows a normal distribution curve with five regions of equal probability. It can be seen, that the region near the mean is more finely subdivided than the areas at the extremes and the normally distributed points are the mid-points between the area bounds. Typically the extreme points are ±3σ (standard deviations) from the mean (nominal) value.

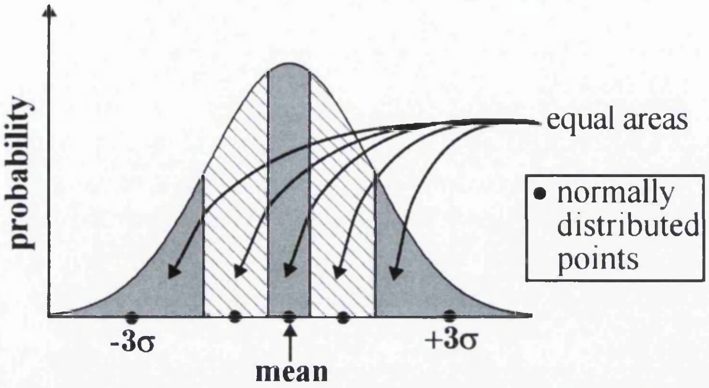


Figure 4.4: Normal distribution of points

The equation of the normal distribution curve is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}, \tag{4.1}$$

where σ is the standard deviation and μ is the mean.

In order to determine the regions of equal probability it is necessary to integrate Equation 4.1 between a lower bound, a, and the upper bound, b, such that

$$\int_a^b f(x)\; dx = A_r. \tag{4.2}$$

The area A_r and the lower bound a are known and the upper bound b is unknown. The solution of Equation 4.2 is non-trivial since there is no exact analytical solution. One way is to use the tables available in the literature or to solve the problem using a numerical technique such as Simpson's rule. The former choice is cumbersome and accuracy is difficult to achieve, and so the latter approach has been adopted in this work. The algorithm implemented in this work to determine the upper and lower bounds of the regions is as follows:

1. Calculate the total area to the left of the current point, e.g. for the first point of 99 points the area = 1/99 = 0.0101.
2. Integrate Equation 4.2 from –4 (say) and increment x by a small amount (say 0.00001) until the solution equals the desired area, e.g. for the first point an area of 0.0101 is desired, this is achieved by numerically integrating Equation 4.2

until the area =0.0101; at this point, the value of x is taken as the upper bound x_1 (or b in Equation 4.2). This can be seen below in Figure 4.5

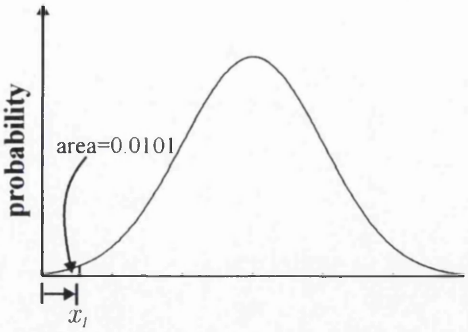


Figure 4.5: Schematic image for determining the upper bound of the integral in Equation 4.2

3. Repeat step 2 from $i = 1$ to $i = n - 1$, where n is the number of sample points. This gives a vector of the bounds $x = \{x_1, x_2, \ldots, x_{n-1}\}$.
4. Determine the mid point between the upper and lower region bounds. In this implementation, the lower bound for the first region takes a value such that the mid-point of the region is three standard deviations from the mean. The lower bound for each subsequent region is the upper bound from the previous region. The upper bound for the last region again takes a value such that the mid-point of the region is three standard deviations from the mean. The corresponding points distribution for a sample of size 99 is shown in Figure 4.6.

An alternative method for applying a normal distribution to the points suggested by Fonseca [4.9] would be to use the uniformly distributed points and apply weightings to the points according to the probability distribution, in this case any type of distribution could be applied using the same set of sample points.

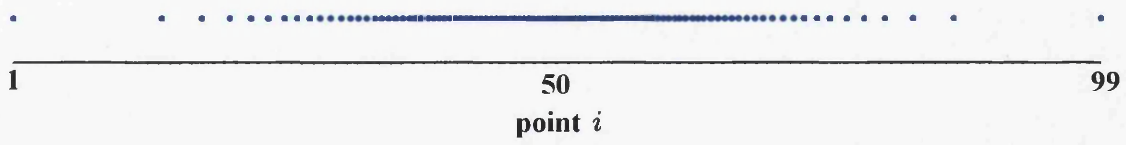


Figure 4.6: 99 normally distributed points

4.2.2 MODELLING

Figure 4.7 shows a typical structure for a deterministic optimization problem, where brainstorming and preliminary studies are carried out to determine the response and control factors. These are then passed to the optimizer, which uses the simulation code to determine the response and hence the next set of design variables. The robust design method has an additional set of input noise factors, these are identified from brainstorming and preliminary work and are then accounted for in the optimization loop. This can be seen schematically in Figure 4.8.

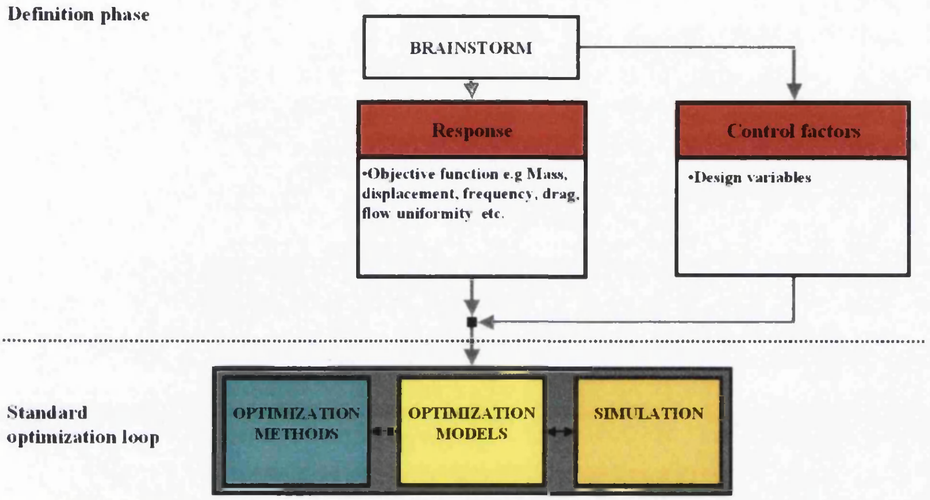


Figure 4.7: Schematic overview of a typical deterministic optimization structure with control factors only

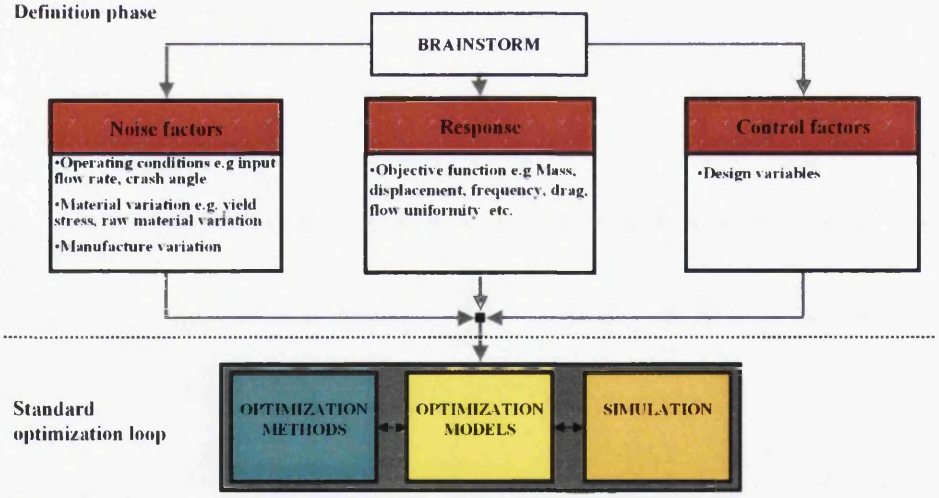

Definition phase
BRAINSTORM
Noise factors
•Operating conditions e.g input flow rate, crash angle
•Material variation e.g. yield stress, raw material variation
•Manufacture variation
Response
•Objective function e.g Mass, displacement, frequency, drag, flow uniformity etc.
Control factors
•Design variables
Standard optimization loop
OPTIMIZATION METHODS
OPTIMIZATION MODELS
SIMULATION

Figure 4.8: Schematic overview of a typical robust design optimization structure with control factors and the addition of noise factors

4.3 Robust design methods

The robust design methods all originate from Taguchi's idea of identifying noise and control factors and then modifying the control factors, such that a product's sensitivity to the variation in the noise factors is minimized for a target performance. The original method was developed with physical experimentation in mind and with the massive increase in the use of computer simulation in place of physical experimentation the robust design methods have evolved and the original methodology has been advanced for use with computer-based simulations. What remains the same is, that the aim of RDO is to generate a design that is minimally sensitive to variation in the noise factors for a target performance. This section gives an overview of the Taguchi approach to robust design and the method's limitations as observed by several authors. Following on, the robust concept exploration method developed by Chen [4.10] is described and the RDO implementation in this work is described. Finally, a review of some of the methods and applications of RDO is given.

4.3.1 TAGUCHI ROBUST DESIGN METHODOLOGY

Typically, a design process consists of 'system' design followed by 'tolerance' design, whereby a product is designed to meet the requirements using scientific and technical knowledge. Then, the manufacturing tolerances are tightened to ensure that a design remains functional. Typically tightening tolerances leads to higher cost [4.11] due to the need for greater manufacturing accuracy.

Since the late 1950's Taguchi has developed methods of design optimization referred to as robust design. The robust design method provides a systematic and efficient approach for finding the near optimum combination of design parameters so that the product is functional, exhibits a high level of performance, and is robust to noise factors. [4.11] "Taguchi's view of optimisation is to reduce process variability and to include noise variables in the experimental design."[4.12] The methods were developed for physical off-line experimentation. A large number of literature sources on Taguchi's robust design and its applications exists, e.g. Fowlkes and Creveling [4.6], Vardeman *et al.* [4.13], Unal and Dean [4.14], Phadke [4.11], American Supplier Institute [4.15], Roy [4.16, 4.17].

Taguchi suggests three major steps in robust design:

1. System design
2. Parameter design
3. Tolerance design

It can be seen, that the steps include system and tolerance design discussed above, but also include a new step called *parameter design*. This is the basis of Taguchi's robust design method. Phadke [4.11] lists eight steps for a robust design procedure:

(i) Identify the main function.
(ii) Identify the noise factors and testing conditions.
(iii) Identify the quality characteristics (objective function) to be optimized.
(iv) Identify the control factors and their alternative levels.
(v) Design the matrix experiment and define the data analysis procedure.
(vi) Conduct the matrix experiment.
(vii) Analyse the data and determine near optimum levels for the control factors.
(viii) Predict the performance at these levels.

These eight steps make the robust design cycle. The first five steps are used for planning the experiment. In the sixth step, the experiment is conducted. In steps seven and eight, the experimental results are analysed and verified.

4.3.1.1 PARAMETER DESIGN

Taguchi introduced the idea of parameter design whereby controllable and uncontrollable factors are firstly identified. Then, according to a performance characteristic, the controllable factors are optimized whilst ensuring that the performance is minimally sensitive to variation caused by the uncontrollable factors. DoE (specifically orthogonal arrays) is used to specify the values of the control factors and the uncontrollable factors. Then the performance characteristic is determined at these specified levels. The DoE is then used to select the optimum levels for the controllable factors such that the system is functional, exhibits a high level of performance under a wide range of conditions, and is robust to uncontrollable factors. Taguchi uses a two-part orthogonal array where an orthogonal array of control variables is crossed with an orthogonal array of noise variables. This produces a design of experiments in which every control variable value is experimented against every noise variable value. For example [4.12], using two two-level noise variables and two two-level control variables, the control variables are the inner array and the noise variables are the outer array. Figure 4.9 shows the resulting crossed-array, 16-experiment design.

The corners of the inner array represent (–1, –1), (–1, +1), (+1, –1), and (+1, +1) for the control variables. The dots in the outer arrays represent the locations of the observations.

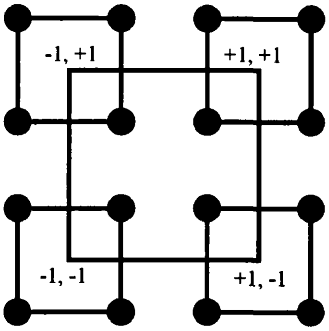


Figure 4.9: Taguchi robust design method: $2^2 \times 2^2$ **crossed array [4.12]**

Taguchi summarizes the mean and variance in each of the outer arrays using a signal-to-noise ratio. So, for the example shown in Figure 4.9, after running the 16 experiments a signal-to-noise ratio (SNR) is given to each outer-array. Then, depending on the aim of the experiment, a different formulation of the SNR is used. In general there are three different aims for the value of the response function: smaller-the-better, larger-the-better or target-is-best, see Taguchi and Phadke [4.18] or Myers and Montgomery [4.12] for further information.

4.3.1.2 Limitations of the Taguchi approach

The Taguchi method uses "statistically designed (physical) experiments in off-line situations where the settings of the noise variables are controlled and systematically introduced and their relationships with design factors studied. Frequently, however, it is too expensive or not practically feasible to control the noise variables, even in off-line experiments. Variations in the noise variables can then invalidate the usual methods of analysis."[4.19]. Freeny and Nair [4.19] develop alternative methods of analysis for situations where the noise variables are uncontrolled but can be observed. "Taguchi's approach appears (to us anyway) to be overcomplicated for simple deterministic models and too expensive for realistic problems" Sacks and Welch [4.7]. Sacks and Welch [4.7] use Latin Hypercube designs (Chapter 3) for computer-based experimentation, they "do not see how a Taguchi-style experimental plan crossing two orthogonal arrays could achieve the same goals without far more experimental effort."

Nair [4.7] in 1992 summarizes a panel discussion on the Taguchi approach, between many of the key researchers in the field including Phadke, Taguchi, Shoemaker, Tsui, Box, Lorenzen, Kacker, Wu, Nelder, Sacks, Welch, Lucas, Myers and Vining. Myers and Montgomery [4.12], Chen *et al.* [4.20, 4.21] discuss in detail the limitations of the Taguchi approach; a summary of them is as follows:

- The use of SNR can lead to misleading results. Myers and Montgomery [4.12] show how for the same value of the SNR there can be a variety of performance distributions.
- The use of a single performance metric, i.e. SNR, is limiting. Chen *et al.* [4.21] state that there are multiple objectives to be satisfied in design. From this, it follows that there must be multiple aspects to quality.
- Many of the DoEs suggested by Taguchi do not allow for estimation of interaction among the control variables.
- The inner and outer arrays used are either saturated or near saturated and are quite economical. However, the crossing of the designs does not produce an economical design. More efficient and simpler experiments and methods of analysis are available.
- Chen *et al.* [4.21] have found it difficult to use the Taguchi approach to find experimental points in the design space where all the engineering constraints are satisfied.
- The Taguchi method will not yield an accurate solution for design problems that embody highly non-linear behaviour.

4.3.2 METHODS AND APPLICATIONS

The robust concept exploration method (RCEM) was developed by Chen [4.10] and has been further developed and applied in many research papers, including Chen *et al.* [4.21, 4.22, 4.23, 4.24, 4.20], Varadarajan *et al.* [4.25], Simpson *et al.* [4.26]. RCEM is an intuitive, practical computer-based approach for solving RDO problems; as such it is the basis used for the RDO procedure implemented in this work (Section 4.4). The following gives an overview of the method, adapted from Ref. [4.26]. Figure 4.10 gives a flow chart of the RCEM procedure.

Step 1: Classify Design Parameters – (box A) This step follows the Taguchi idea of identifying the noise and control factors and the responses, and combining them using a 'p-diagram' Phadke [4.11].

Step 2: Screening experiments – (boxes B, C and D) DoE is used with a simulation to perform initial screening experiments to identify and eliminate the factors, which have little or no influence on the design, and to also reduce the design space.

Step 3: Generate a response surface model – (boxes B, C and E) DoE and the 'expensive' simulation is used to generate response surface models, which are functions of both control and noise factors, to replace the original (computationally expensive) analysis programs.

Step 4: Determine the values of the control factors – (boxes A, E and F) This step is essentially the formulation of the multi-objective problem, and solving it using an appropriate method. This is done using the response surface model in a framework of a "compromise decision support problem" [4.27] to determine the values of the control factors, to achieve a performance which is as close as possible to the target value and to minimize variations around the target.

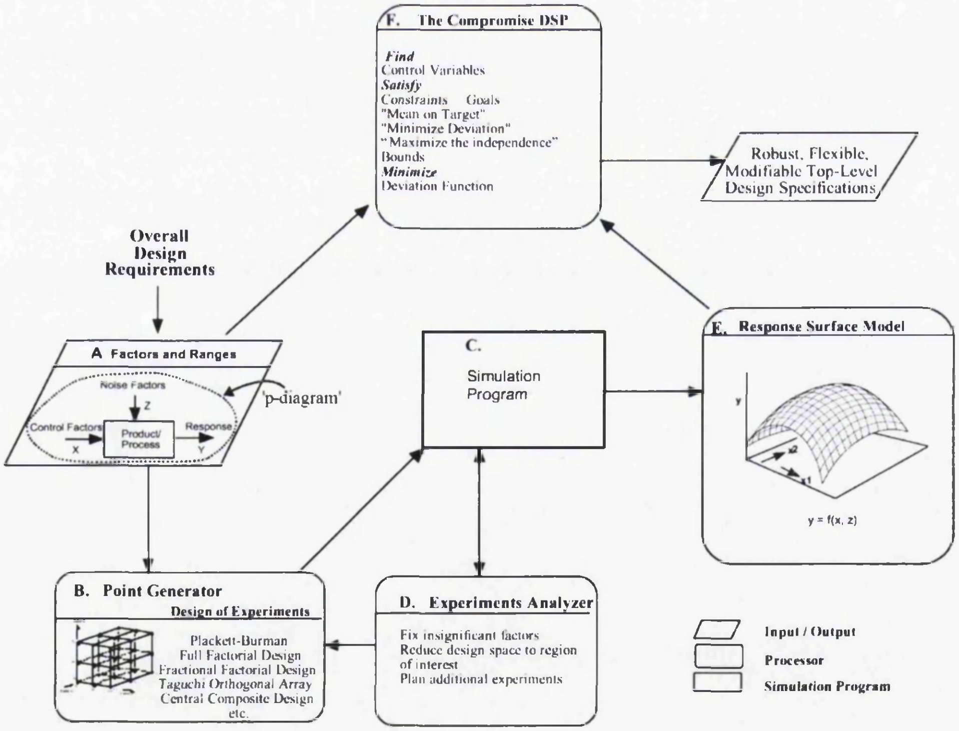


Figure 4.10: Computer infrastructure of RCEM [4.23]

Sacks and Welch [4.7] discuss their approach of building approximation functions relating each response to all input parameters and then optimize via these approximations rather than directly through the computationally expensive computer model. Identifying that "the optimization clearly depends on the accuracies of the approximating models", and that, "when factor ranges are sufficiently narrow, we have found second-order polynomial models to give enough accuracy. When the factors have wide ranges, however, leading to complex input-output relationships, or when data are scarce, the interpolators described by Currin *et al.* [4.28], Sacks *et al.* [4.29, 4.30], and Welch *et al.* [4.31] are more flexible and data-adaptive and tend to be more accurate and successful for prediction and optimization."

Sundaresan [4.32] introduced a single objective function with a sensitivity index for use in a RDO loop. The single objective function incorporates both performance and variation by means of the addition of a sensitivity index to the original objective function [4.33]. The overriding problem with this is the need to use weighting factors to generate the trade-off between the mean and variance, and as such leads to the problems discussed in Section 3.5.1.

Engel and Huele [4.34] use a "Statistical Robust Design" method that is based on Taguchi's method and extends it by developing a response surface model at the nominal (controllable and uncontrollable factor) settings, called the "conditional response surface", and using this model random variables are substituted for the noise factors. In this manner process mean and process variance can be obtained. Engel and Huele (1996) discuss second order response models in the noise factors.

Lautenschlager and Eschenauer [4.35] use response surfaces for the robust design which account for variation in both the design variables and other system parameters. The results showed that robustness needs to be considered during optimization and that using a combined array is superior to Taguchi's crossed-array approach. As observed by Chen *et al.* [4.20], Welch *et al.* [4.36] propose combining control and noise factors into a single array thus modelling the response rather than expected loss, and approximating a prediction model for loss based on the fitted-response model. Shoemaker *et al.* [4.37] develop this further. These proposed modifications to the Taguchi method, however, still involve a single performance metric.

Lee and Park [4.38] suggested a technique for robust design, whereby an optimum value insensitive to variations of the design variables within a feasible region is sought using a multi-objective function of the mean and variance, and a penalty applied to the constraints. They also assumed a tolerance of three standard deviations on the design variables (this is also assumed in the slit die RDO problem in Chapter 5).

In Chapter 5 and the publications of Bates *et al.* [4.39, 4.40, 4.41] robust design is applied to slit die extrusion accounting for noise in the design variables and the other important system parameters. In the approach described in Section 4.4, the solver is not replaced by a response surface, instead the uncertainty of the response evaluations due to variations in the noise factors is generated by sampling around each point specified by the optimizer using an optimal Latin Hypercube technique, which is a derivative of the Monte Carlo method (Section 4.5.2).

Venter *et al.* [4.42] propose an approach to robust design using particle swarm and genetic algorithm optimization. One of the problems with using gradient-free algorithms is the associated high computation cost. It is impractical to use gradient-free methods to solve problems under uncertainty, without "specifically addressing the resulting high-computational cost" [4.42]. Venter *et al.* [4.42] account for uncertainty in the design variables using the function evaluations at the design points sampled in a typical optimization run. So, "little or no additional computational cost is required". The response evaluations at the design points are used to construct response surface approximations to estimate the sensitivity of the response functions to uncertainty in the design variables; however, the approach cannot account for uncertainty in the uncontrollable factors. The aim of the optimization is to "steer clear of designs that exhibit high sensitivity to uncertainty".

Hersleth *et al.* [4.43] who look at the effect of contextual factors on liking wine using the classical Taguchi robust design approach. The aim of the study was to try and determine which type of wine 'taste rating' was least sensitive to the context in which it was served. The noise factors were the surroundings in which the wine was served and whether or not food was provided during the tasting. The control factors were three processes that affected the taste and the body of the wine.

Su and Renaud [4.33] give a review of some other RDO methods.

Du and Chen [4.5] look at 'feasibility robustness', i.e. robust design to ensure that the design achieves robust design objectives but also remains feasible. They analyse and compare several existing feasibility modelling techniques in robust design. The authors identify that, according to Parkinson *et al.* [4.44], a design is described to have feasibility robustness, if it can be characterized by a definable probability, set by designers, to remain feasible relative to the nominal constraint boundaries as it undergoes variations. Du and Chen [4.5] use a 'most probable point based importance sampling technique' for the problem. They state that "no matter what objective expression we use to achieve the robustness of product performance, it is even more critical to maintain the design feasibility under variations".

4.4 ROBUST DESIGN IMPLEMENTATION

In order to develop a procedure for RDO, the slit die design example that is described in detail in Chapter 5 is used as a 'test bed'. The simulator called POLYSIM requires less than 30 seconds CPU time on a standard PC to calculate the response characteristic for a given set of control and noise factors. As such, it is possible to use the analysis code to directly compute the response instead of using an approximation of the design space as suggested by RCEM. Applying the procedure, using more computationally expensive simulations, may be prohibitive and an accurate design space approximation would be required. Varadarajan *et al.* [4.25] compare various different approximation techniques in a RCEM process such as RSM and ANN.

The implementation of a RDO procedure in this work is similar to the intuitive RCEM described above. The method accounts for both variation in the design variables and also variation in the other system parameters. It is summarized in Figures 4.11 and 4.12 and is as follows.

The basis of the procedure is a deterministic optimization loop, whereby the optimizer only sees a single objective function, and it only specifies the nominal values of the control factors. On top of this, there are a series of procedures. At each design point specified by the optimizer a set of nominal values of the control factors is available. At this point in the design space a sample of size n is taken. In the current implementation the sample is generated using an optimal Latin Hypercube sample (described in Chapter 3), but any sampling technique may be used. The advantage of the optimal Latin Hypercube is that it is a space-filling sampling technique and as such, covers the full range of all of the factors. The sample encompasses variations in the manufacture (in the slit die example it is the shape of the choker bar), variations in the operating conditions and variations in the raw material etc. The variations of these uncontrollable factors are defined *a priori*, for example from physical testing it is observed that the manufacturing variation or tolerance is 0.1mm and the variation is normally distributed about the nominal value. The response is calculated using the simulator at each sample point, resulting in n response values from which the mean and standard deviation of the sample is calculated. The simulations can be run independently, so if there are n processors available then n samples can be run in parallel. This effectively means that there is negligible difference in the wall-clock time required to calculate the deterministic response value compared to the stochastic response value, although the number of processors required increases by a factor of n.

The RDO problem is bi-objective since there are two response values to be optimized, i.e. the mean and the standard deviation. In the case of the slit die it is desired that the mean and the standard deviation are both minimized simultaneously. Minimizing competing objectives is not usually possible and instead a trade-off between them is required. In order to determine the trade-off, a multi-objective technique is used to generate a Pareto set of solutions (discussed in Chapter 3). In Chapter 5 three multi-objective techniques are compared, these are NBI, PP and PFM-PP; they are discussed in Chapter 3. Das [4.45] also uses the NBI method to carry out robustness optimization.

In the current implementation it is necessary to carry out a separate optimization run for each point on the Pareto front, i.e. if 30 points are desired then 30 optimization runs are needed, although techniques can be used to reduce the number of iterations required by the optimizer. These are discussed in Chapter 3. Using a multi-objective optimization method means that the two responses (mean and standard deviation) are combined into a single objective function value; this value is passed to the deterministic optimizer and it is the only value it sees. Based on this single value, the optimizer then specifies the next design variable values and the process is repeated. The final outcome of the RDO calculations is a Pareto-set of solutions from which the 'robust design' is chosen. Zhang *et al.* [4.46] (extending the work of Chen *et al.* [4.47]) present a procedure for deriving an approximation of the Pareto efficient frontier to explore alternative robust design solutions. It eliminates the needs of solving the original bi-objective optimization problem repeatedly as is done in the RDO procedure implemented here.

The procedure and the RDO structure implemented in this work are summarized below in Figure 4.11 and 4.12. The robust design is indicated in Figure 4.11 as RD. The reason for this being the robust solution is that it has a low mean and a lower performance variation than say the solution with the lowest mean. Although the minimum performance isn't as good, the overall performance is more predictable and the mean is still acceptable. Appendix A2 gives a detailed description of the computer implementation of the procedure.

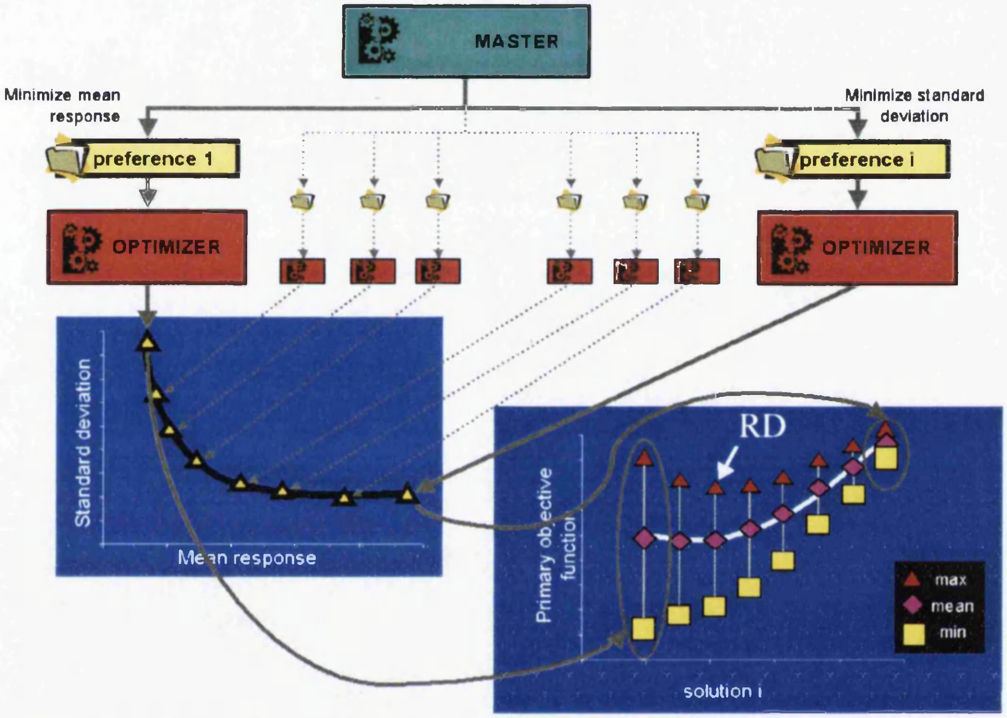


Figure 4.11: Schematic view of the implemented RDO procedure

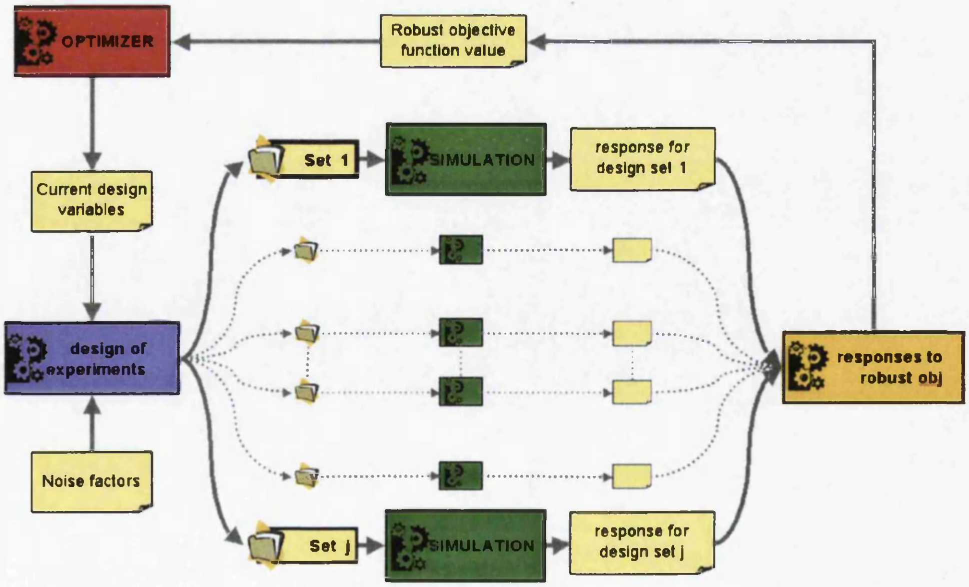


Figure 4.12: Calculation of the robust design objective function

The additional computational cost compared to a deterministic optimization run is summarized as follows:

- At each design point a sample of size n is computed; this requires n-times extra CPU time, if the response evaluations are calculated serially using a single CPU then the wall-clock time is also n-times greater. However, if the simulations are run in parallel then there is no additional wall-clock time required.
- To generate k points on the Pareto front, then at most k additional optimization runs are required using the same optimization method. The extra computational effort depends on the multi-objective method used. In the case of the slit die RDO problem in Chapter 5, three multi-objective methods were compared and it was seen that using the same optimization method, i.e. DOT, a 10-times reduction in the function evaluations was achieved using different multi-objective methods.

4.5 SOME OTHER 'DESIGN FOR UNCERTAINTY' METHODS

4.5.1 COMPUTATIONAL STOCHASTIC MECHANICS

Computational stochastic mechanics encompasses methods for modelling uncertainties in loading, material and geometric properties of structures using probability and statistics. Schuëller [4.48, 4.49] gives a comprehensive review on the state-of-the-art of computational stochastic mechanics; see also e.g. Fonseca *et al.* [4.50]. Methods and issues involved in computational stochastic mechanics are

- the Monte Carlo simulation method and improving its efficiency;
- representations of stochastic processes, i.e. identifying the distributions of random parameters;
- representation of stochastic processes using approximation models;
- procedures for predicting structural response and hence structural reliability;
- methods for estimating response variability such as perturbation techniques.

4.5.1.1 MONTE CARLO SIMULATION

Monte Carlo simulation was named after the city of Monte Carlo, Monaco, where the primary attractions are casinos containing games involving probability. It is a statistical method for simulating a model, by repeatedly and randomly generating values for system variables that have a known range and probability distribution, hence an uncertain value at any particular time. The Monte Carlo simulation is a simple method. The number of simulations required is independent of the number of variables unlike some DoE sampling methods. It is computer intensive, but each simulation can be run totally independently so that it can be completely parallelized to match the number of CPU available.

From the literature it is evident that articles written by Marczyk have the same overriding theme: uncertainty must be accounted for in any CAE design process, especially if it is entails designing for chaotic situations such as automotive crash, and Monte Carlo simulation must be used to solve uncertainty problems. Numerous and passionate arguments are provided in Refs. [4.51, 4.52, 4.53, 4.54] for example:

- "Chaos is a deterministic phenomenon; it implies unpredictability but not randomness. What characterizes chaos is extreme sensitivity to initial conditions…crash possesses a clear chaotic flavour...many manufacturers still resist to recognize that crash is a stochastic and chaotic phenomenon and therefore should be treated as such."
- "In systems dominated and characterized by a mixture of chaotic and random behaviour, talking of optimization is simply absurd. You can't optimize a car for crash when you don't know what other car, or obstacle, it will hit, with which velocity, angle, mass, number of passengers etc. Because of tolerances in manufacturing and assembly, it is even impossible to manufacture two identical cars. So what can we optimize? … Since initial conditions are never known exactly, it makes no sense to speak of optimization of chaotic systems…each crash phenomenon is unique. Global patterns are all we can ever hope to comprehend. This is clearly alien territory for optimization…better crashworthiness designs can be obtained if the problem is approached with the Monte Carlo method, the natural method for crash."
- "According to Galois [4.55], 'If you put tomfoolery into a computer, nothing comes out but tomfoolery. But this tomfoolery, having passed through a very expensive machine, is somehow ennobled and no one dares criticize it.' "

Reuter and Hülsmann [4.56] uses a 'stochastic improvement method' based on Monte-Carlo simulation, the method they use is available in the commercial package ST-ORM (Section 4.6) and is also known as 'return mapping'[4.57]. In their method a 'cloud of points' is used instead of a response surface; instead of using derivatives to move along a response surface, a correlation matrix is used to show how the parameters of a system influence each other. An optimization method *is not* specifically used but improvement directions are identified using the response values from the cloud of points. To generate the cloud of points, the probability distributions and the allowable ranges of the design variables is defined together with the probability distributions of the other noise factors, then a Monte Carlo simulation is run using the solver. For each sample the distance to the target or objective is calculated and the sample that is closest to the target is identified. Then, using the values of the design variables for the closest sample, the probability distributions of the design variables are modified such that the mean values are equal to the closest sample. This process of generating a cloud of point is then repeated until the distance to a target value is less than some specified value. The target can be either some performance characteristic or could be to match the physical test to the simulation model – this is known as stochastic validation. Reuter and Hülsmann [4.56] use the method for designing a driver airbag system. Reuter *et al.* [4.2] use the method for head impact simulation in a crash and state that the method is suitable only for cases where relatively small improvements are necessary to meet the requirements.

It can be seen, that there is a similarity between the method described in the previous paragraph and the robust design method implemented in this work (described in Section 4.4). (i) In both methods a cloud of points is sampled around a point in the design space and the cloud represents a variation of the performance caused by imparting a probability distribution on the system parameters and design variables; (ii) Changing the design variables using optimization in the latter approach is equivalent to altering the mean values of the design variables in the former approach. The methods differ on how the information is used. For the former approach a single target is specified *a priori* whereas in the latter approach the aim is to generate a number of Pareto solutions representing the trade-off between the mean and variation of performance.

4.5.1.2 PERTURBATION TECHNIQUES

Perturbation techniques are non-statistical methods used for generating first or second order response surface models of response variation. They are often used in reliability analysis for calculating the sensitivity of the response with respect to the design variables and in robust design for the sensitivities of the mean and variance of the response with respect to the design variables. The advantages are the method's tractability and the use of the approximation saves CPU time [4.49]. The disadvantages are that it is required that the random variables involved in the analysis do not deviate much from their expected values.

For example, Doltsinis and Kang [4.3] and Kang *et al.* [4.58] proposed a method of robust design using a second order perturbation technique [4.59]. In their approach, the approximations were used to calculate the sensitivities of the mean and variance of the response with respect to the design variables (the random variation of factors other than the design variables was not considered) and then optimization was used to minimize the two objectives. The trade-off between the objective functions was represented using Pareto optimality and the results were verified using Monte Carlo simulation, where the distribution of the stochastic input parameters was assumed to be normal.

4.5.1.3 RELIABILITY-BASED DESIGN OPTIMIZATION

Structural reliability analysis is an analytical method for determining the probability that a design will not violate limits that define failure. "The integration of reliability analysis into design optimization represents the reliability-based design optimization (RBDO) model." Kharmanda *et al.* [4.60]. "The applicability of RBDO relies on the availability of the precise distribution of stochastic parameters."[4.3]

The difference between robust design and RBDO is, according to Zang [4.61], that in robust design a design is sought that is relatively insensitive to small changes in the uncertain quantities; in RBDO a design is sought that has a probability of failure that is less than some acceptable (invariably small) value. The classification of the two methods is shown in Figure 4.13 according to Hyuse [4.62] and Zang *et al.* [4.61, 4.63].

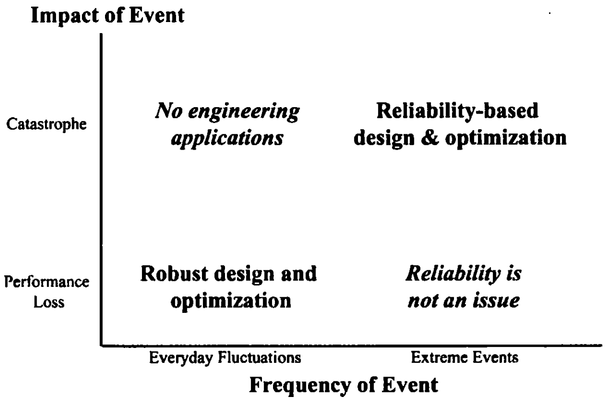

Impact of Event
Catastrophe
No engineering applications
Reliability-based design & optimization
Performance Loss
Robust design and optimization
Reliability is not an issue
Everyday Fluctuations
Extreme Events
Frequency of Event

Figure 4.13: Classification of robust design optimization and reliability based design optimization [4.61, 4.63, 4.62]

Frangopol and Maute [4.64] state that optimization methodologies can be classified into two groups: robust design optimization (RDO) and reliability-based design optimization (RBDO). They state that RDO methods are based "purely on deterministic analysis and attempt to maximize the deterministic performance and simultaneously to minimize the sensitivity of the performance with respect to random parameters. This approach leads to a multi-objective optimization problem capturing the impact of uncertainties only in a qualitative sense. Unlike RDO approaches, RBDO methods allow the design for a specific risk and target reliability level accounting for the various sources of uncertainty in a quantitative sense. RBDO approaches are based on stochastic analysis methods and, therefore, algorithmic-wise more challenging and computationally more expensive compared to deterministic approaches."

RBDO methods are based on the assumption that the design space is divided into two regions: success and failure. [4.63] The objective is to find the best design, which is located at a point in the design space away from the failure region such, that the probability of failure is acceptable. By defining this failure region the problem becomes constrained and is solved using standard optimization methods. Zang *et al.* [4.63] identify the major difficulties of RBDO stating that "it can be computationally expensive to calculate the probability of failure and that the probability of failure constraint can be a highly non-linear function." In most cases, the probability of failure cannot be evaluated by analytical means and numerical methods are used such as [4.64]:

- Monte Carlo simulation – these methods feature generality, simplicity, and effectiveness on problems that are highly non-linear with respect to the uncertainty parameters. The most serious drawback is the computational costs, in particular when the reliability level is high, that is the failure probability low.
- Perturbation techniques or First- and second-order reliability methods (FORM/SORM)- these methods are used to approximate the reliability index. These methods search for the most probable point (MPP) on the failure using linear or quadratic approximations.

Traditional RBDO requires a double loop iteration process, e.g. see Figure 4.14. The inner loop is to find the most probable point (MPP) (see for example, Chen and Du [4.65]) and the outer loop is to optimize the RBDO problem with reliability objectives or constraints [4.66]. Some single-loop methods have been developed which are typically less computationally intensive than the double-loop approach, e.g. Du and Chen [4.67].

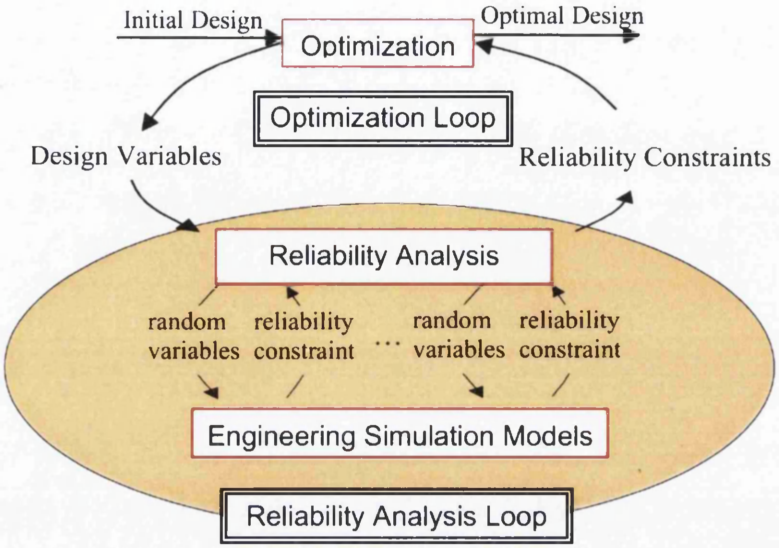


Figure 4.14: The double-loop reliability-based design [4.68]

Kharmanda *et al.* [4.69] introduce a method to simultaneously solve the reliability and optimization problem in a 'hybrid design space'. Further details and current work on RBDO can be found in Kharmanda [4.69], Yang *et al.* [4.70], Choi and Youn [4.71], Yang and Gu [4.66], Koch and Kodiyalam [4.72], Cullimore [4.73]. See for a good overview of these methods. Zang *et al.* [4.63], Langley [4.74], Grooteman [4.75], Frangopol and Maute [4.64] give a comprehensive overview of all of the stochastic methods.

4.5.2 SIX-SIGMA

Six-sigma is essentially a quality management process aimed at ensuring that a product will fall within ± six standard deviations within the specification, i.e. there are less than 3.4 defect products per million (4.5 standard deviations shifted by 1.5 standard deviations from the mean). Motorola [4.76] pioneered the six-sigma idea in the 1980s for manufacturing quality. "One of Motorola's most significant contributions was to change the discussion of quality from one where quality levels were measured in percentages (parts per hundred) to a discussion of parts per million or even parts per billion. Motorola correctly pointed out that modern technology was so complex that old ideas about acceptable quality levels were no longer acceptable."[4.77]

"Embarking on a Six Sigma program means delivering top-quality service and products while virtually eliminating all internal inefficiencies."[4.78] Although the process was implemented to improve manufacturing quality it can also be applied to other aspects of a business such as "optimizing response time to inquiries, maximizing the speed and accuracy with which inventory and materials are supplied, and fool-proofing such support processes from errors, inaccuracies and inefficiency." "It is a technique used to establish a common goal for all employees in an organization: reduce variability (i.e., standard deviation) in everything they do." "In a Six Sigma organization, employees assess their job functions with respect to how they improve the organization. They define their goals, or the ideal of excellence in their roles, and quantify where they are currently with respect to these ideals. Then they work to minimize the gap and achieve Six Sigma by a certain date." "When properly implemented, six-sigma reduces inefficiencies and produces very high yields and returns. This requires proper planning and implementation." [4.78] "Unlike other quality programs, such as total quality management, six-sigma is a disciplined system of using extremely rigorous data-gathering and statistical analysis to pinpoint sources of errors and ways of eliminating them." [4.79]

There are five steps involved in a six-sigma process [4.80]

(i) Define – Identify areas for improvement that are critical to quality.

(ii) Measure – Measure the defects related to the process.

(iii) Analyze – Identify parameters most responsible for quality loss. This involves carrying out DoE or data analysis.

(iv) Improve – Optimize the process.

(v) Control – structure the process to reduce variance using quality engineering methods such as robust design or reliability based optimization.

A precise description of the method is difficult to give since it is more of a framework describing the procedures for using other established methods to achieve a set of goals. Indeed, Stamatis [4.81] talks about how "we shouldn't jump on the bandwagon with six sigma" and that "six sigma is a repackaging of existing ideas". "One of the more common criticisms is that it has little to offer that can't be found elsewhere, that it's simply a marketing ploy." [4.82]

The following is adapted from Ref [4.78]. In a six sigma process the process potential index C_p, and the process capability index C_{pk}, are used in a 'total process characterization' scheme. C_p is a measure of a process's potential capability and is

$$C_p = \frac{USL - LSL}{6\sigma_{physical}}, \tag{4.3}$$

where $USL - LSL$ is the allowable variation, i.e. the upper specification limit minus the lower specification limit, $6\sigma_{physical}$ is the actual variation (standard deviation of the physical data of a process multiplied by six). Therefore an increase in C_p means a decrease in the standard deviation and the process is less variable. A value of $C_p < 1.0$ means the process is 'potentially incapable' of meeting the specification limits. Conversely, $C_p \geq 1.0$ the process has the potential of being capable of meeting the specification limits. However, a high C_p value doesn't guarantee that a production process falls within specification limits because the C_p value doesn't imply that the actual spread coincides with the allowable spread (i.e., the specification limits). This is why the C_p is called the process potential. C_{pk} is the measure of a process's ability to create a product within specification limits, it is defined as

$$C_{pk} = \frac{\mu - SL_{closest}}{3\sigma}, \tag{4.4}$$

where C_{pk} represents the difference between the actual process average μ and the closest specification limit, $SL_{closest}$, divided by three standard deviations. A value of $C_{pk} < 1.0$ means the process is 'incapable' of producing a product within specification limits. Conversely, when a process, $C_{pk} \geq 1.0$ the process is capable of producing a product within specification limits.

Overall, the objective is to maximize C_p and C_{pk}, and in order for a process to be classified as six-sigma then $C_p \geq 2.0$ and $C_{pk} \geq 2.0$.

Perez-Wilson [4.78], Deshpande [4.83], Chase [4.84], NCAT [4.85] and Pyzdek [4.77] give further details on six-sigma. Ramberg [4.82] discusses the criticisms of six-sigma. Walmsley [4.79] gives an overview of some of the companies using six-sigma and the benefits that have been achieved.

4.6 CONCLUDING REMARKS

In this chapter several methods for designing under uncertainty have been discussed. Particular attention has been paid to computationally based robust design optimization and some other techniques have been described.

In spite of Marczyk's continued discussions on why the use of "optimization is absurd", it is generally recognized that optimization can be used for stochastic problems and it needs to account for uncertainty inherent in many design problems. Marczyk's arguments seem to be based purely on the assumption that optimization can and is only used for problems under ideal deterministic conditions. It has been shown, that the stochastic nature of problems is being accounted for in optimization problems. Currently, the use of Monte Carlo simulation using the expensive solver is still unrealistic, especially in a commercial environment. However, the use of meta-models of the design space to replace the expensive solver is feasible and is being used. The robust design method implemented in this work uses a variation of the Monte Carlo method, i.e. the optimal Latin Hypercube, inside an optimization loop. The solver can be anything representative of the problem's performance, e.g. the expensive solver, a low-fidelity model of the problem or a response surface model. In any case the choice of simulation is and always will be dependent on the amount of computational power available.

4.7 REFERENCES

[4.1] Trosset M.W., Alexandrov N.M., Watson L.T., "New Methods for Robust Design Using Computer Simulation", Spring Research Conference, June 2003.

[4.2]Reuter R., Hoffmann R., Kamarajan J., "Application of Stochastic Simulation in the Automotive Industry" AMERI-PAM 2000 Southfield, Michigan, October 2000.

[4.3] Doltsinis I., Kang Z., "Robust design optimization of structures with stochastic parameters", proc. Fifth World Congress on Computational Mechanics (WCCM V), July 7-12, Vienna, Austria, 2002.

[4.4] Taguchi G., "Introduction to Quality Engineering: Designing Quality into Products and Processes", Kraus International, New York, 1986.

[4.5] Du X. and Chen W., "Towards a Better Understanding of Modeling Feasibility Robustness in Engineering Design", ASME J. Mech. Design, 122(4), 385-394, 2000.

[4.6] Fowlkes W.Y., and Creveling C. M., "Engineering Methods for Robust Product Design: Using Taguchi Methods in Technology and Product Development", Addison Wesley Longman, Inc., ISBN: 0-201-63367-1, 1st edition, 1995.

[4.7] Nair V.N., "Taguchi's Parameter Design: A Panel Discussion", Technometrics, Vol. 34, No. 2, pp. 127-161, 1992.

[4.8] Welch W.J., Sacks J., "A system for quality improvement via computer experiments", Communications in Statistics Theory and Methods, 20:477-495, 1991.

[4.9] Personal communication with José Fonseca, University of Wales Swansea, 2003.

[4.10] Chen, W. "Robust Conceptual Exploration for Configuring Complex Engineering Systems," Ph.D. Dissertation, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, August 1995.

[4.11] Phadke M.S., "Quality Engineering using robust design", New York, N.J.: Prentice Hall, ISBN: 0137451679,1989.

[4.12] Myers R.H., Montgomery D.C., "Response surface methodology: Process and Product Optimization Using Designed Experiments", John Wiley & Sons, New York, NY, 1976.

[4.13] Vardeman S., Brenner L., Wulf W., "Robust Product Design" February 11, IE 361, 2000. http://www.public.iastate.edu/~vardeman/IE361/s00mini/brenner.htm

[4.14] Unal, R. and E. B. Dean "Design for Cost and Quality: The Robust Design Approach," Journal of Parametrics, August, Vol. XI, No. 1, pp. 73-93, 1991.

[4.15] American Supplier Institute Inc, "Taguchi Methods: Implementation Manual", ASI, Dearborn, MI, 1989.

[4.16] Roy R.K., "16 steps to improvement, A well-run DOE method leads the way to better products and processes." Quality Magazine, June 2001.
http://www.qualitydigest.com/june01/html/sixteen.html

[4.17] Roy R.K., Design of Experiments Using the Taguchi Approach: 16 Steps to Product and Process Improvement. New York: John Wiley & Sons, 2001.

[4.18] Taguchi G., Phadke M.S., "Quality Engineering Through Design Optimization," in Conference Record, GLOBECOM 84 Meeting, IEEE Communications Society (Atlanta, GA, November), pp. 1106-1113, 1984.

[4.19] Freeny A.E., and Nair V.N., "Robust Parameter Design with Uncontrolled Noise Variables", AT&T Bell Laboratories, Murray Hill, NJ, 1992.

[4.20] Chen, W., Allen, J. K., Tsui, K.-L. and Mistree, F., "A Procedure for Robust Design: Minimizing Variations Caused by Noise and Control Factors," Journal of Mechanical Design, Vol. 118, No. 4, pp. 478-485, 1996.

[4.21] Chen, W., Allen, J. K., Mavris, D., and Mistree, F. "A Concept Exploration Method for Determining Robust Top-Level Specifications" Engineering Optimization, 26, 137-158, 1996.

[4.22] Chen W., Garimella R., Michelena N., "Robust design for improved vehicle handling under a range of maneuver conditions" Proceedings of The 1999 ASME Design Engineering Technical Conferences September 12-15, 1999, Las Vegas, Nevada

[4.23] Chen W., Allen J.K., Mistree F., "The Robust Concept Exploration Method for Enhancing Concurrent Systems Design", Journal of Concurrent Engineering: Research and Applications, 5(3), 203–217, 1997.

[4.24] Chen W., Fu W., Biggers S., Latour R., "An Affordable Approach for Robust Design of Thick Laminated Composite Structure", Optimization and Engineering, 1(3), 305-322, 2000.

[4.25] Varadarajan, S., Chen, W., and Pelka, C., "Robust Concept Exploration Method of Propulsion Systems with Enhanced Model Approximation Capabilities", Engineering Optimization, 32(3), 309-334, 2000.

[4.26] Simpson T.W., Chen W., Allen J.K., Mistree F. "Conceptual Design of a Family of Products Through the Use of the Robust Concept Exploration Method", AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, Washington, September 4-6, 1996.

[4.27] Mistree F., Hughes O. F., Bras B. A., "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm", Structural Optimization: Status and Promise, AIAA, Washington, D.C., pp. 247-289, 1993.

[4.28] Currin C., Mitchell T., Morris M., Ylvisaker D., "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments", Journal of the American Statistical Association, 86, pp. 953-963, 1991.

[4.29] Sacks J., Schiller S.B., Welch W.J., "Designs for computer experiments" Technometrics, Vol. 31 Issue 1, pp41-48, Feb 1989.

[4.30] Sacks J., Welch W. J., Mitchell T. J., Wynn H. P., "Design and Analysis of Computer Experiments (with discussion)", Statistical Science, 4, pp. 409-435, 1989.

[4.31] Welch W. J., Buck R. J., Sacks J., Wynn H. P., Mitchell T. J., Morris M. D., "Screening, Predicting, and Computer Experiments," Technometrics, 34, 15-25, 1992.

[4.32] Sundaresan S., Ishii K., Houser D.R., "A Robust Optimization Procedure with Variations on Design Variables and Constraints," Advances in Design Automation, ASME DE-Vol. 69-1, pp. 379-386, 1993.

[4.33] Su J., Renaud J.E., "Automatic Differentiation in Robust Optimization," AIAA Journal, Vol. 35, No. 6, pp. 1072, 1997.

[4.34] Engel J., Huele A.F., "A Generalized Linear Modelling approach to robust design", Technometrics, vol. 38, no. 4, pp. 365-373, November 1996.

[4.35] Lautenschlager U., Eschenauer H. A., "Improved Robust Design Approximations applied to Multiobjective Shape Optimization Problems", 51-AAM2-2, proc. 3rd WCSMO, Buffalo, NY, 1999.

[4.36] Welch W.J., Yu T.K., Kang S.M. Wu J., "Computer Experiments for Quality Control by Parameter Design", Quality Technology, Vol. 22, pp. 15-22, 1990.

[4.37] Shoemaker A.C., Tsui K.-L., Wu J., "Economical Experimentation Methods for Robust Design", Technometrics, Vol. 33, No. 4, pp. 415-427. 1991.

[4.38] Lee K.H., Park G.J., "Robust optimization considering tolerances of design variables", Computers & Structures, Volume 79, Issue 1, Pages 77-86., January 2001

[4.39] Bates S.J., Sienz J., Pittman J.F.T., "Robust Design Optimization of Extrusion Slit Die Design Using Several Multi-Objective Optimization Methods", proc. 5th WCSMO conf., Venice, Italy, 2003.

[4.40] Bates S.J., Sienz J., Pittman J.F.T., Langley D.S. "Application of Robust Design Optimization to Extrusion Slit Die Design", proc. 3rd Int. conf. on Engineering Computational Technology, Prague, ISBN: 0-948749-86-5, 2002.

[4.41] Bates S.J., Sienz J., Pittman J.F.T. "Comparison of conventional optimization with robust design techniques for slit die design", Proc. 4th ASMO UK / ISSMO conf., Newcastle, UK, ISBN: 0-7017-0137-4, 2002.

[4.42] Venter G., Haftka R.T., Sobieski J.S., "Robust Design Using Particle Swarm and Genetic Algorithm Optimization", WCSMO 5, 2003.

[4.43] Hersleth M., Mevik B.-H., Næs T., Guinard J.-X., "Effect of contextual factors on liking for wine-use of robust design methodology", Food Quality and Preference, Volume 14, Issue 7, pp 615-622, October 2003.

[4.44] Parkinson A., Sorensen C., Pourhassan N., A General Approach for Robust Optimal Design," Transactions of the ASME, Vol. 115, pp.74-80, 1993.

[4.45] Das I., "Robustness Optimization for Constrained Nonlinear Programming Problems" Engineering Optimization, vol. 32, no. 5, pp. 585-618, June 2000.

[4.46] Zhang J., Wiecek M.M., Chen W., "Local approximation of the efficient frontier in robust design", 1999 ASME design engineering technical conferences, Las Vegas, Nevada. DETC99/DAC-8566, September 12-15, 1999.

[4.47] Che W., Wiecek, M., Zhang J., "Quality Utility: A Compromise Programming Approach to Robust Design", accepted by ASME J. of Mech. Design, Sept 1998.

[4.48] Schuëller G.I., "Computational stochastic mechanics – recent advances", Computers & Structures, Volume 79, Issues 22-25, pp 2225-2234, September 2001.

[4.49] Schuëller G.I., "A State-of-the-Art Report on Computational Stochastic Mechanics", Probabilistic Engineering Mechanics, 12(4), pp. 197-321, 4 Oct. 1997.

[4.50] Fonseca J., Mares C, Friswell M.I., Mottershead J.E., "The Propagation of Parameter Uncertainty through Structural Dynamics Models", in Proceedings of IMAC-XXI, Kissimmee, Florida, February 3-6, 2003.

[4.51] J Marczyk J., "Putting The Uncertainty Back into CAE", Desktop Engineering online, http://www.deskeng.com , March 2002.

[4.52] Marcyk J. "Automotive crash simulation: a personal perspective" Easi Engineering, January 2001.

[4.53] Hargreaves B., "Chaos theory", Vehicle Engineering & Design, pp.23, May 2002.

[4.54] Marczyk J. "Stochastic Multidisciplinary Improvement: Beyond Optimization" AIAA-2000-4929, 2000.

[4.55] Shannon, "The Apt Quotation - Pierre Galois", N.Y. Times, at C11, col. 1, June 12, 1990.

[4.56] Reuter R., Hülsmann J., "Achieving Design Targets through Stochastic Simulation", MADYMO users' conference, Paris 2000.

[4.57] Doltsinis I., Rau F., Werner M., "Analysis of random systems", In: Doltsinis I. (Ed): Stochastic analysis of multivariate systems in computational mechanics and engineering, CIMNE, Barcelona, 1999.

[4.58] Kang Z., Doltsinis I., Cheng G., Gu Y., "Robust design optimization of structures with thermal stress using perturbation-based stochastic finite element method", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.127-128, 2003.

[4.59] Liu W.K., Belytschko A., "Random field finite element", Int. J. for Numerical methods in Engineering, 23:1831-1845, 1988.

[4.60] Kharmanda G., Mohsine A., El-Hami A., "Efficient reliability-based design optimization for dynamic strcutures", WCSMO 5, Lido di Jesolo, Italy, May 19-23, pp.135-136, 2003.

[4.61] Zang T.A., "Risk-based design overview", ASCAC methods development peer review, Nov 27-29, 2001.

[4.62] Huyse L., "Solving Problems of Optimization Under Uncertainty as Statistical Decision Problems", AIAA-2001-1519, 2001.

[4.63] Zang T.A., Hemsch M.J., Hilburger M.W., Kenny S.P., Luckring J.M., Maghami P., Padula S.L., Stroud W.J., "Needs and Opportunities for Uncertainty-Based Multidisciplinary Design Methods for Aerospace Vehicles" NASA/TM-2002-211462, Langley Research Center, Hampton, Virginia, 2002.

[4.64] Frangopol, D.M.; Maute, K. (2003):Life-cycle reliability-based optimization of civil and aerospace structures. Computers & Structures, Vol. 81, 397-410.

[4.65] Chen, W., and Du X., "A Most Probable Point Based Approach for Efficient Uncertainty Analysis", ASME Design Automation Conference, Paper No. DAC14263, Baltimore, MA, September 10-13, 2000.

[4.66] Yang R. J., Gu L., "Experience with Approximate Reliability-Based Optimization Methods", AIAA-2003-1781, 2003.

[4.67] Du X., Chen W., "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design", ASME Design Engineering Technical Conferences, DETC2002/DAC-34127, Montreal, Canada, 2002.

[4.68] Du X., Sudjianto A., "Reliability-based design with the mixture of random and interval variables", DETC2003/DAC-48709, Chicago, Illinois USA, 2003.

[4.69] Kharmanda G., Mohamed A., Lemaire M., "Efficient reliability-based design optimization using a hybrid space with application to finite element analysis" Struc. Multidisc. Opt., 24 (3), pp. 233–245, Sep. 2002.

[4.70] Yang R. J., Gu L., Tho C. H., Choi K. K., Youn B. D., "Reliability-based Multidisciplinary design optimization of a full vehicle system", AIAA-2002-1758, 2002.

[4.71] Choi K.K., Youn B.D., "Hybrid analysis method for reliability based design optimization", proceedings of ASME design Engineering technical conferences, Pittsburgh, September 2001.

[4.72] Koch P.N., and Kodiyalam S. "Variable complexity structural reliability analysis for efficient reliability-based design optimization" AIAA-99-1210, 1999.

[4.73] Brent A. Cullimore "Reliability Engineering and Robust Design: New Methods for Thermal/Fluid Engineering", C&R White Paper, Revision 2, May 15, 2000.

[4.74] Langley R.S. "Unified Approach to Probabilistic and Possibilistic Analysis of Uncertain Systems". J. Eng. Mech., vol. 126, no. 11, pp. 1163–1172, 2000.

[4.75] Grooteman F. P., "Advanced Stochastic Method of Probabilistic Analysis", NLR-TP-99398, 1999.

[4.76] http://www.motorola.com/

[4.77] Pyzdek, T., The Complete Guide to Six Sigma, Quality Press, 2000.

[4.78] Perez-Wilson M., "Six Sigma Strategies: Creating Excellence in the Workplace" Quality Magazine, December 1997.
http://www.qualitydigest.com/dec97/html/sixsigma.html

[4.79] Walmsley A. "Six Sigma Enigma", Globe and Mail Report on Business Magazine, November 2001. http://www.ge.com/investor/article/

[4.80] Sheh M., "CAE IN Design for SixSigma (DFSS): An Emerging Paradigm" Engineous Software Inc. HP CAE Technology Seminar June 2001"
http://www.hp.co.kr/event/mcae/pdf/T2-3.pdf

[4.81] Stamatis D.H. "Who needs six sigma anyway?", Quality Magazine, May 2000.
http://www.qualitydigest.com/may00/html/sixsigmacon.html

[4.82] Ramberg J.S."Six Sigma: Fad or Fundamental?" Quality Magazine, May 2000.
http://www.qualitydigest.com/may00/html/sixsigmapro.html

[4.83] Deshpande P.B. "Emerging technologies and six sigma" Simulation and Advanced Controls, Inc., and University of Louisville, Louisville, Kentucky 77(4), April 1998.

[4.84] Chase N., "Six Sigma Black Belts Stamp Out Tough Quality", Quality Magazine, August 1999. http://www.qualitymag.com/articles/1999/aug99/0899f3.html

[4.85] National Center for Advanced Technologies "Analyzing Risk through the Application of Six Sigma Concepts" http://www.ar.navy.mil/reflib/mod6.pdf

SECTION 2

INDUSTRIAL APPLICATIONS

CHAPTER 5

SLIT DIE EXTRUSION DESIGN

Summary: This chapter uses a 'black-box' computer simulation of the performance of a 1.2m wide slit die, which is used to produce 1.9mm thick sheets of plastic. The simulation takes into account the coupling of melt flow and die body deflection due to melt pressure. One way of achieving uniform flow from a slit die, and hence uniformly thick sheets, is to use a restricting mechanism called a choker bar, and to determine its shape requires optimization. In the first part, deterministic optimization is applied to determine the choker bar shapes required for three materials of varying degrees of shear thinning at two flow rates and under ideal conditions. Three methods are used: DOT, MARS and GA. Overall, GA finds better solutions but requires 10-times more function evaluations than DOT or MARS. In the second part, robust design optimization is used to determine choker bar shapes for one material at one flow rate assuming real conditions. Three multi-objective optimization methods are used to generate the Pareto set of solutions, which represent the trade-off between the mean and variation of performance: the physical programming method (PPM), the normal boundary intersection method (NBI) and a method introduced in this work called the 'Pareto front marching method' using physical programming (PFM-PP). For the implementations used and for the current problem, the Pareto front marching method generates a uniform distribution of points resulting in a smooth Pareto front and requires five-times fewer iterations to generate the solutions than the two other methods. Comparison is made between the solutions generated using a uniformly distributed sample of size 10, with those gained using a normally distributed sample of size 100.

5.1 INTRODUCTION

This chapter is set out as follows:

Section 5.1: Gives an introduction to the problem and a review of previous work on die design and in particular optimization of slit dies.

Section 5.2: Gives a brief description of the computer simulation used to predict the slit die performance.

Section 5.3: Describes the choker bar design problem. Firstly, the parameterization of the choker bar is explained. The materials and processing conditions are then identified. Next, the question of whether the determined optimum profiles can be achieved by bending the restrictor bar is considered.

Section 5.4: Deterministic optimization is used to determine the optimum choker bar profiles that give the best possible exit flow uniformity for the extrusion of three materials with different degrees of shear thinning, each at two flow rates and under **ideal** conditions.

Section 5.5: Robust design optimization, using a uniformly distributed sample of size 10, is applied to determine choker bar profiles to give optimum melt flow distribution under **real** conditions for the material Montecatini A5 at one flow rate. Concentrating on the ability and the speed of generating a uniformly distributed Pareto set for the bi-objective problem, this section compares the performance of the physical programming method, the normal boundary intersection method and a method introduced here called the 'Pareto front marching method' using physical programming.

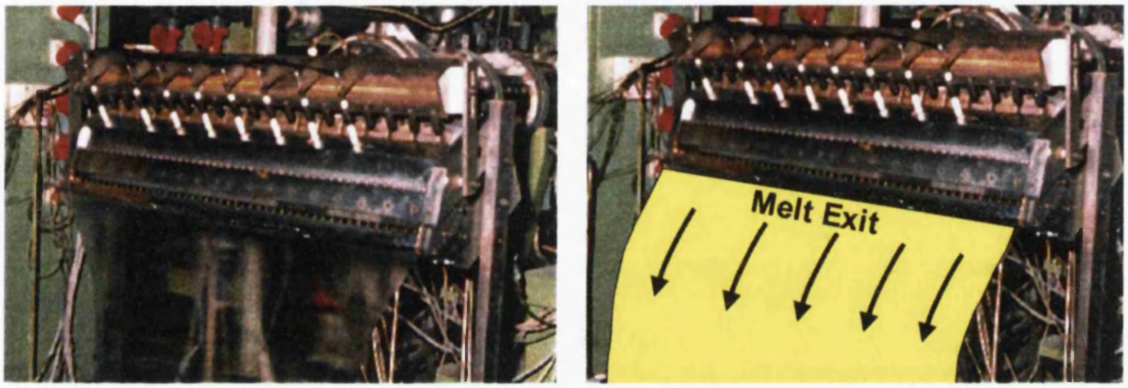
Section 5.6 This section is a continuation of Section 5.5; a normally distributed sample of size 100 and the PFM-PP method is used for the robust design optimization. Comparison of the results is made with those of Section 5.5.

Section 5.7: Gives some concluding remarks.

Section 5.8: Lists the references used in the chapter.

5.1.1 Overview

Figure 5.1(a) shows a photograph of a 1.2m wide slit die extruder, typically used to manufacture 1.9mm thick sheets of plastic. This is shown schematically in Figure 5.1(b), where the yellow region is a sketch of the sheet exiting the slit die in the direction indicated. Figure 5.2 is a plan-view cross-section of the die showing that the melt flow enters through a circular feed channel and flows through a series of regions in the distribution channel until it reaches the exit.



(a) (b)

Figure 5.1: (a) Photograph of a 1.2m wide slit die; (b) schematic view of the melt exiting the slit die

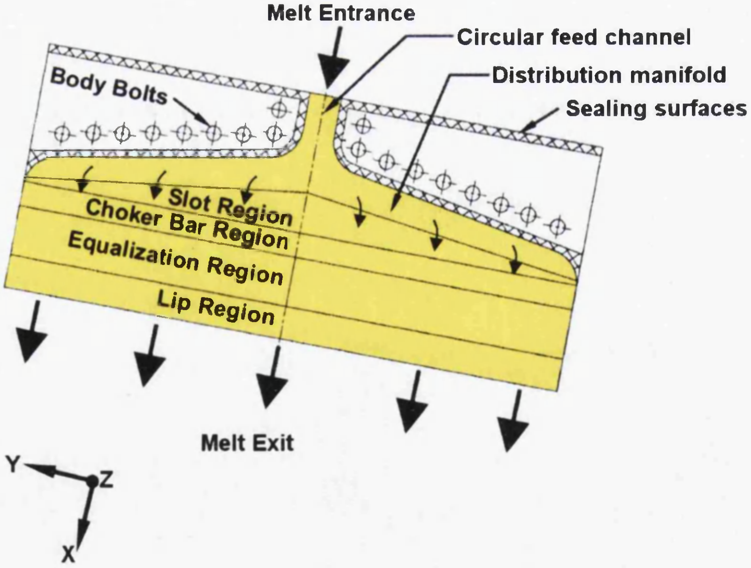


Figure 5.2: Cross-sectional plan view of the melt flow through the slit die

An important manufacturing requirement is to produce sheets or films of uniform thickness. To achieve this requires the design of a slit die with a uniform flow rate across the die exit. There are several methods to achieve an optimum (ideally uniform) flow rate distribution across the die exit, the method used in this chapter is to use a flow restricting mechanism called a 'choker bar'. Figure 5.3 shows the die cross-section along the principal flow direction and the location of the choker bar. Figure 5.4 shows a sketch of the choker bar with non-uniform flow distribution before the choker bar, and the ideal situation of uniform flow after. The choker bar has a curved profile, restricting flow in some areas more than in others, so the objective of designing it is to determine the shape of its profile.

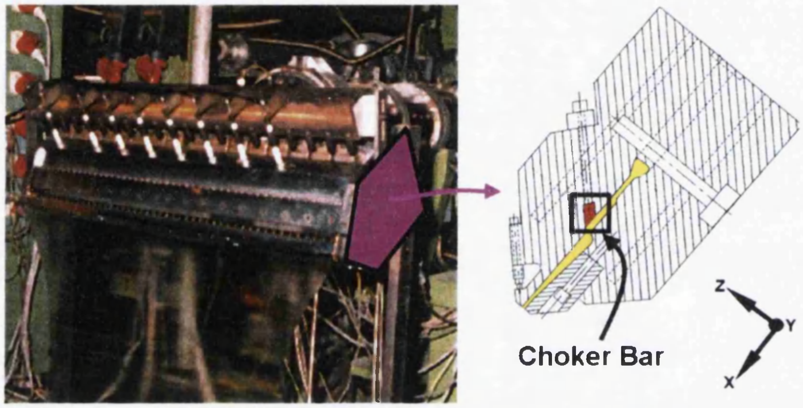


Figure 5.3: Die cross-sectional side view showing choker bar location

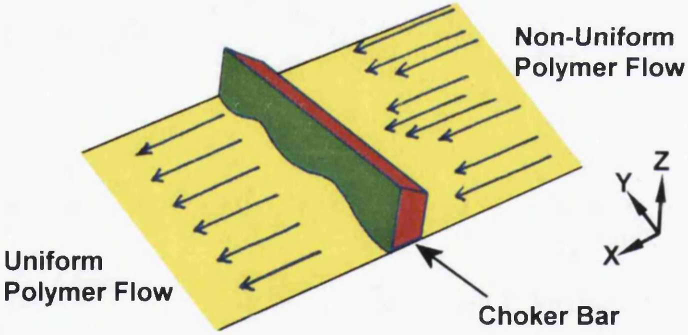


Figure 5.4: Schematic diagram of a choker bar altering the melt flow distribution from a non-uniform distribution to a (ideally) uniform distribution

In this chapter the following questions are asked: what is the best possible choker bar profile in certain specific applications, and how effective is this in bringing about the desired uniform flow distribution?

5.1.2 BACKGROUND

This section is adapted from [5.1] and is here for completeness.

As stated in Section 5.1.1, an important requirement in the design of slit dies is to achieve a uniform flow rate across the die exit, to give a sheet or film of uniform thickness. Although it is theoretically possible to design dies where the flow distribution is independent of flow properties (specifically the degree of shear thinning), uniform distribution may not be achieved in practice. Winter and Fritz [5.2] presented a theory for the design of coat-hanger dies, with circular or rectangular section distribution manifolds. For the latter, provided the aspect ratio of the manifold is at least 10, the theory predicts material independence of the flow distribution. For wide dies, however, this leads to dies that are long in the flow direction. The loads produced by the melt pressure can then cause significant deflection, opening up the melt flow channel, particularly at downstream positions near the centre. Flow here is then higher than predicted theoretically and the required uniform distribution is not achieved. Another feature of designs according to this theory is that the distribution manifold curves down, and at its lower end forms part of the lips at the edge of the die. Numerical simulations [5.3] show that though melt distribution is excellent across most of the die exit (in the absence of deflection), flow rate falls off in the edge region where the manifold joins the lip. Wortberg and Tempeler [5.4] proposed an alternative approach using a so-called representative viscosity and requiring uniform shear rate, and therefore uniform viscosity, throughout the manifold and slit. Difficulties in manufacturing such dies have, however, limited the application of this principle.

The performance of slit dies is therefore, in practice, dependent both on material flow properties and operating conditions, and because dies are often used for a variety of extrusion tasks, adjustable flow restrictors are introduced with the objective of achieving the required uniform flow distribution. These can be in the form of either a membrane [5.5] or a conventional ‘choker bar’ as described in Section 5.1.1. Both require on-line adjustment, either manually by trial and error, or automatically by feedback to servos. In any given case, it is generally not possible to predict what adjustments are required, and a run-in time is required to set up the required restrictor settings. It is also not known in advance of extrusion trials whether uniform flow distribution can indeed be achieved using the available restrictor adjustments. If the optimum settings could be predicted, the run-in time could be eliminated or reduced, reducing scrap and lost production time. Additionally, simulations of the die performance would show, in advance of trials, whether use of the optimized settings would actually achieve an acceptable flow distribution for a given application.

Work on the numerical simulation of slit die performance up to the mid-1990s was reviewed by Sander and Pittman in Ref. [5.6]. In that paper, for the first time, the interaction of melt flow and die body deflection was taken into account in a fully coupled analysis, and predictions were confirmed in experiments on a 1.2m wide high pressure die. Melt flow was treated using the generalized Hele-Shaw (HS) approximation, and die deflection using Mindlin thick plate theory, reducing both analyses to two dimensions and allowing rapid solutions on a personal computer. Subsequently, Gifford [5.7] developed a coupled analysis using three-dimensional finite elements. Comparisons with two-dimensional analyses or experimental results were not presented, so the advantages provided by this significantly larger computation are unclear.

Whilst software of this type provides tools for predicting the performance of a given design, determination of the 'best' design is still difficult. Using optimization can help the designer with this further step. Due to the relatively simple geometry, work so far published has been mainly concerned with flat dies. This is reviewed below.

Smith, Tucker and Tortorelli [5.8, 5.9] modelled Newtonian and non-Newtonian isothermal flow in a coat hanger die using a finite element (FE) implementation of the generalized HS two-dimensional approximation, and optimized the die to minimize pressure drop subject to exit flow uniformity being within a set tolerance. The sensitivities needed for the gradient-based optimizer were calculated using the adjoint method. In a first example, a design, including a conventional flow distribution channel, was optimized using 19 geometrical design variables and one constraint. In a second example, the flow channel height was allowed to vary freely without a predetermined distribution channel. The geometry was defined using 516 geometrical variables and the objective was to minimize the exit velocity variation subject to 506 height gradient constraints. The increased degrees of freedom resulted in a design with a lower pressure drop. The same authors give a fuller account of their work in [5.10] and [5.11] and extend it in a number of ways. Sensitivity calculations by direct differentiation and adjoint methods are compared, and simultaneous minimization of residence time variations is added. They note that the resulting dies are long, with high pressure-drops and susceptibility to 'clam shelling' (die body deflection as described above). Su Yeon Na and Tai-Yong Lee [5.12] used three-dimensional FE analysis of Newtonian isothermal flow in a coat-hanger die. Design variables defined the width of the distribution manifold as a function of lateral position, and the objective function was defined in terms of the uniformity of the flow direction pressure gradient at the die exit. A penalty function was introduced to impose smoothness of the manifold height profile. The quadratic optimization problem required multiple solutions on a Cray supercomputer. Yeh Wang [5.13] used the one-dimensional lubrication approximation for isothermal power law flow in a coat-hanger die with Taguchi and Simplex optimization methods. Design variables controlled the radius of the distribution manifold as a function of transverse position, for a fixed drop angle of the manifold. The objective was to minimize a uniformity index defined using local flow rates at a number of predefined sections at the die exit or to minimize the pressure drop across the die. The effect of varying the drop angle was investigated, but from comparisons with three-dimensional FE analysis, it was concluded that the results based on the one-dimensional approximation were misleading. These design exercises have been for a specific material extruded under specified processing conditions. Furthermore, no die deflection has been taken into account.

As described in Section 5.1.1, the present work is the application of optimization to choker bar design. The simulation code in the optimization procedure is the coupled two-dimensional analysis of die performance mentioned above [5.6]; thus the effects of die body deflection are taken into account. In the optimization procedure, repeated analyses are required, and the computational economies of the two-dimensional formulation make it feasible to enclose the analysis within an optimization loop.

A number of aspects of melt flow in the die are not modelled in the work. Non-isothermal conditions are excluded. These are generally undesirable in the operation of slit dies, and in [5.14] it was shown that melt temperature variations are usually small, and means to minimize them were proposed. Based upon the research in Ref. [5.14] a variation of ± 10°C is used in this work. It is recognized that wall slip can occur for certain polymers, but this is not included here, though the HS formulation could be extended to include it. Viscoelastic effects may be significant in some cases. Flow in parallel slits will be close to viscometric, with kinematics and pressure drops well predicted by generalized Newtonian models, but elasticity will result in additional pressure drops where flow converges into narrow slits, such as at the exit from the distribution manifold or under a flow restrictor. Intrinsically, the HS formulation does not model converging flows, and a three-dimensional analysis would be required.

Here, a numerical simulation (described in Section 5.2) of an existing die design is used to investigate extrusion for a range of materials under varying process conditions (described in Section 5.3) taking die body deflection into account.

5.2 SLIT DIE SIMULATION

A computer simulation of the slit die performance is used in the optimization procedure. The software was developed in the research of Sander [5.15]. It is called 'POLYSIM' and its formulation is described below for completeness and is used as a 'black box'.

POLYSIM is an FE analysis that takes into account the coupling of melt flow and die body deflection due to melt pressure to give predictions of the dynamic operating characteristics of slit dies. The melt flow is modelled using a HS formulation, assuming isothermal flow (which was justified in a further study of thermal effects in dies [5.6]), and generalized Newtonian flow behaviour. The simplifications result in a non-linear potential equation for pressure on the plane of the flow slit, which combines momentum and mass conservation,

$$\frac{\partial}{\partial x}\left[S \frac{\partial p}{\partial x} \right] + \frac{\partial}{\partial y}\left[S \frac{\partial p}{\partial y} \right] = Q_0 , \qquad (5.1)$$

where p is the pressure, Q_0 represents a volumetric source term and S is the flow conductivity. For isothermal flow of a fluid that is characterized by the power law model

$$\tau = \left(\frac{\dot{\gamma}}{\phi} \right)^n , \qquad (5.2)$$

where τ is the shear stress, $\dot{\gamma}$ is the shear rate and ϕ is the melt consistency, the flow conductivity may be written as

$$S = \frac{\phi n}{\{ (\partial p / \partial x)^2 + (\partial p / \partial y)^2 \}^{(n-1)/2n}} \frac{(H/2)^{(2n+1)/n}}{2n+1} , \qquad (5.3)$$

where n is the power law index and H is the height of the planar cavity between parallel or nearly parallel plates. Note however, that the overall representation of viscosity uses a logarithmic polynomial in the viscosity. Power law parameters are fitted locally, element by element, for evaluation of Equation 5.3. Gap-wise averaged flow velocities may be recovered from the pressure field by

$$\overline{v}_x = -2\frac{S}{H}\frac{\partial p}{\partial x} \qquad \overline{v}_y = -2\frac{S}{H}\frac{\partial p}{\partial y}, \tag{5.4}$$

and the local flow rates between the plates, per unit width in the coordinate directions, are then

$$Q_x = H\overline{v}_x \qquad Q_y = H\overline{v}_y \ . \tag{5.5}$$

It is apparent from Equations 5.1 and 5.3 that the problem for the pressure field is non-linear, and an iterative solution scheme will be required. The problem is solved using FEs; with the slit height(s) defined at the FE nodes within each element. Step changes in height can occur along element boundaries, and values can vary smoothly within elements. Whilst the HS approximation is valid in the slit, the aspect ratio of the distribution channel is such that it cannot be treated directly in this way. Instead, an equivalent HS channel is defined using results for pressure drops in developed flow through prismatic channels of various cross sections [5.16]

$$H_e = \left\{\frac{D_h}{2}\left[\frac{4(4n+2)}{3n+1}\frac{2AD_h}{W\lambda_a}\right]^n\right\}^{1/(2n+1)}, \tag{5.6}$$

where λ_a is a shape factor appropriate to the manifold cross section, as given by Miller [5.16]. A is the cross sectional area of the manifold, D_h its hydraulic diameter and W its width.

Die deflection is modelled using a Mindlin-Reissner (MR) thick plate FE formulation, which again reduces the problem to two dimensions, with plate thicknesses defined at nodes within each element. The implementation is taken from Hinton and Owen [5.17]. The formulation allows for transverse shear deformation, with the underlying assumptions being: (i) the displacements are small relative to the plate thickness; (ii) the stress normal to the mid-plane surface is negligible; and (iii) normals to the mid-surface before deformation remain straight but not necessarily normal to the mid-surface after deformation. By integrating through the thickness, the formulation provides a two-dimensional approximation, wherein weighted average transverse displacements $w(x,y)$, and rotations normal to the mid-plane, $\phi_x(x,y)$ and $\phi_y(x,y)$, are obtained. The

FE equations are conventionally obtained by minimization of the functional for total potential energy of the MR plate, which may be written as:

$$\begin{aligned}\Pi\left(w, \Psi_x, \Psi_y \right) = {} & \frac{1}{2} \iint\limits_A \varepsilon_b^T D_b \varepsilon_b dxdy + \frac{1}{2} \iint\limits_A \varepsilon_s^T D_s \varepsilon_s dxdy \\ & - \iint\limits_A wq dxdy - \int\limits_{\Gamma_s} \left(M_n \phi_n + M_{ns} \phi_s + Q_n w \right) d\Gamma \end{aligned} \quad . \qquad (5.7)$$

Here, w is the z-direction transverse displacement and Ψ_x and Ψ_y are the rotations about the x and y axes, which lie in the plane of the plate. The normal distributed load per unit area is q, and M_n, M_{ns} and Q_n are the moments and transverse shear forces per unit length of the portion Γ_S of the plate boundary Γ. The bending and shear strains are ε_b and ε_s. The terms in Equation 5.7 are respectively the bending and transverse shear strain energies and the potential energy associated with the lateral loads and the boundary forces. FEs based on MR assumptions have the important advantage that they require only $C(0)$ continuity of the unknowns, unlike elements based on the classical thin plate Kirchhoff theory, which require $C(1)$ continuity. Boundary conditions and constraints are applied to model the effect of the body bolts connecting the die halves. The overall solution procedure, shown in Figure 5.5, uses a segregated approach to the coupled problem.

Starting from the un-deflected geometry of the die, the melt flow is analyzed for the specified material and processing conditions. The resulting melt pressure field provides the pressure loads q on the two die halves. The pressure is interpolated from the FE mesh used for the flow analysis to those used for the plate analyses. Then the deflection of each half is calculated. From these results the flow channel height at each node of the flow mesh is modified and the flow analysis repeated. Iteration between the flow and deflection analyses is continued to convergence, to provide full coupling. The convergence criterion for both the inner HS, Picard iteration and the outer flow-deflection loop requires the magnitude of the fractional change in pressure at every node to be less than 10^{-4}. Convergence of the HS loop typically requires 5 or 6 cycles, and the outer loop 6 to 10.

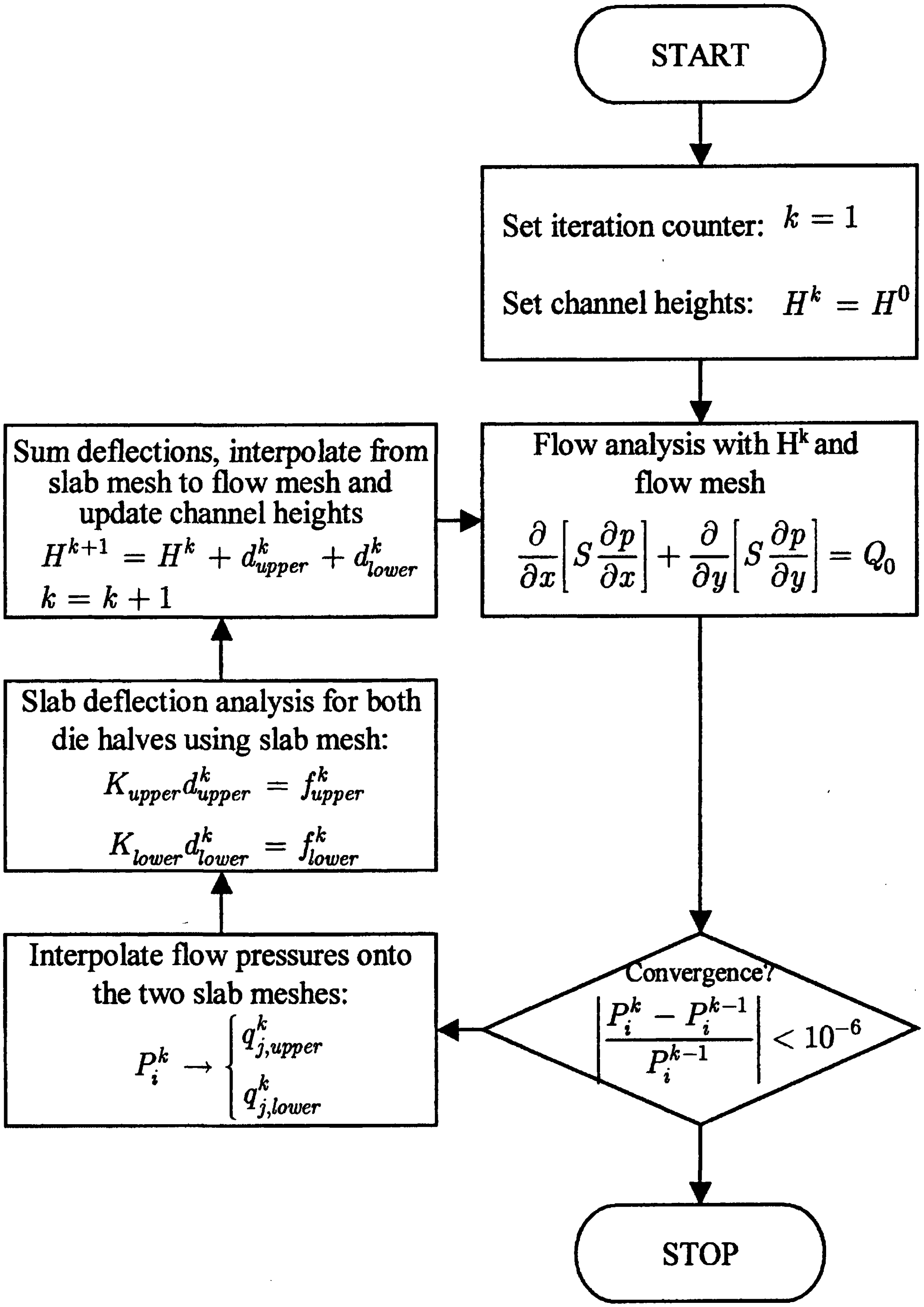

START
Set iteration counter: $k = 1$
Set channel heights: $H^k = H^0$
Flow analysis with H^k and flow mesh
$\frac{\partial}{\partial x}\left[S\frac{\partial p}{\partial x}\right] + \frac{\partial}{\partial y}\left[S\frac{\partial p}{\partial y}\right] = Q_0$
Convergence?
$\left|\frac{P_i^k - P_i^{k-1}}{P_i^{k-1}}\right| < 10^{-6}$
Interpolate flow pressures onto the two slab meshes:
$P_i^k \rightarrow \begin{cases} q_{j,upper}^k \\ q_{j,lower}^k \end{cases}$
Slab deflection analysis for both die halves using slab mesh:
$K_{upper} d_{upper}^k = f_{upper}^k$
$K_{lower} d_{lower}^k = f_{lower}^k$
Sum deflections, interpolate from slab mesh to flow mesh and update channel heights
$H^{k+1} = H^k + d_{upper}^k + d_{lower}^k$
$k = k + 1$
STOP

Figure 5.5: Flow chart for the fluid-structure interaction solution for flow in a defelcted die solved via an iterative loop

The die described in Section 5.1.1 was studied previously in [5.6]. Examples of the results of this simulation are as follows: Figure 5.6 illustrates the importance of die body deflection for this die operating at a pressure drop of 157bar, with a material with a power law index $n = 0.4$. Computed deflection of the upper die half on the centre line 10mm upstream of the exit is 0.409mm, whilst the experimentally measured value [5.18] is 0.418mm, a difference of 0.009mm or 2%. Towards the edge of the die, the deflection is smaller, the computed value being 0.332mm and the experimental 0.270mm. The larger discrepancy at the edge of the die is believed to be due to the constraining effect of the edge seal, which was not modelled. Nevertheless, the agreement between computed and measured pressure drops is within 2%. Neglect of the die deflection increases the computed pressure drop by 25%.

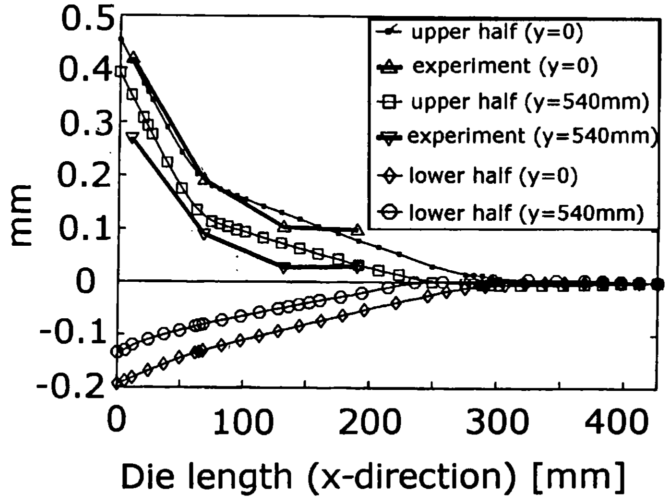


Figure 5.6: Comparison of computed and experimental die deflections on the centre line and close to the edge, for a 1.2m wide die operating with a pressure drop of 157bar. For coordinate system see Figure 5.3

The influence of the die deflection on the flow distribution is illustrated in Figure 5.7. The computed result without consideration of the deflection, and without the use of choker bar or flex lip adjustments (Figure 5.7(a)), gives a ratio of flow rate at the centre to flow rate at the edge of 1.2. Here, the flow rate and the exit velocity curves run parallel, because the lip gap is constant. In the real case, including the deflection, the situation is much worse (Figure 5.7(b)): the simulation shows flow channelled towards the centre, resulting in a ratio of 2.0. Clearly this die is not well suited for extrusion of a material with $n = 0.4$, and it would be difficult to determine, without using optimization, whether uniform flow could be achieved by restrictor adjustment.

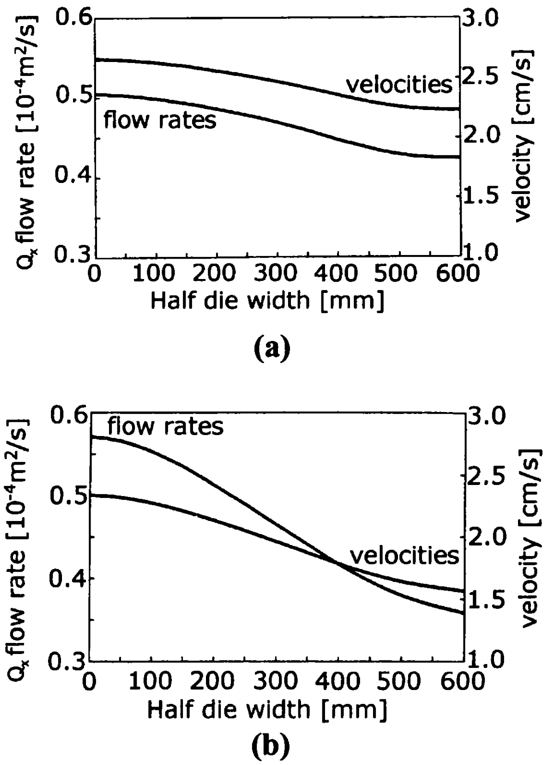


Figure 5.7: Flow distribution at the exit of the 1.2m wide die, for a material with *n* = 0.4, (a) computed without consideration of die body deflection; (b) computed taking die body deflection into account

5.3 Problem description

This section describes the choker bar design problem. Firstly, the parameterization of the choker bar is explained. The materials and processing conditions are then identified. Next, the question of whether the determined optimum profiles can be achieved by bending the restrictor bar is considered.

5.3.1 Parameterization

The nodal channel heights beneath the choker bar represent its profile. These are parameterized using heights at five points equally distributed across the half-width of the die – these being the vector of design variables $\mathbf{d}$ as shown in Figure 5.8 – with interpolation to the nodes of the FE mesh by piecewise cubic splines. Continuity of the second derivative at the ends of the splines ensures smoothness of the profile. Figure 5.8(i) shows the initial choker bar. The optimizer achieves the optimized profile by altering the five shape design variables in Figure 5.8 (ii) to (vi). No constraints on the curvature are applied in the optimization because it is desirable to determine the optimum profiles without these limitations, especially as a choker bar could be machined to obtain the required form.

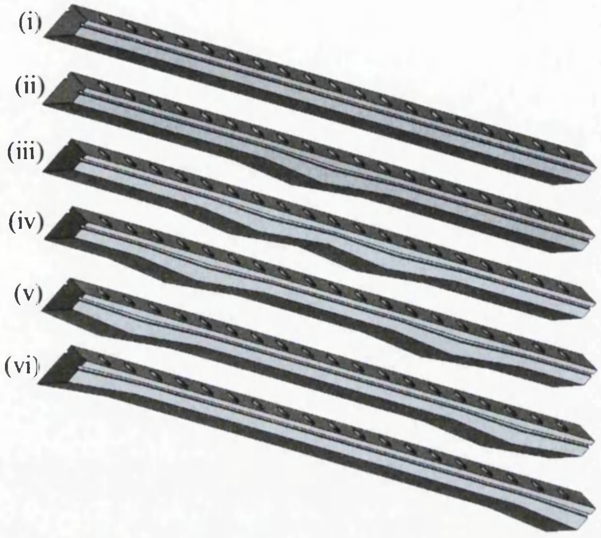


Figure 5.8: Complete choker bar: (i) initial shape; (ii)-(vi) choker bar shape variables 1 to 5 respectively which can vary from 0.1 mm to 3.2 mm

5.3.2 Objective function

Since the function of the choker bar is to produce a uniform flow distribution across the die, the objective function is a positive exit flow uniformity index that goes to zero for perfect uniformity. Other considerations, such as limiting variations in residence time or material deformation are not appropriate, and should instead be considered in optimizing overall die design including the distribution manifold. A suitable dimensionless flow uniformity index can be defined as:

$$F(\mathbf{d}) = \frac{\left(\int_0^L (Q_x - \overline{Q}_x)^2 \, dl \right)^{1/2}}{\overline{Q}_x L^{1/2}} \tag{5.8}$$

where $F(\mathbf{d})$ is the objective function, $\mathbf{d}$ is the design variable vector of flow channel heights below the restrictor, Q_x is exit flow rate in the the local extrusion direction, $\overline{Q}_x$, the target value, is its average value across the half width, L, of the die, and l is the position coordinate along the die exit in the y direction. No constraint on the pressure drop through the die is applied. However, the pressure drop is monitored. The move limits (side constraints) for the design variables are set as 3.2mm (choker bar fully open) and 0.1mm (choker bar almost closed). Thus, the un-constrained optimization problem is defined mathematically as:

$$\min F(\mathbf{d}) \tag{5.9}$$

$$\text{Subject to: } 0.1\text{mm} \leq d_i \leq 3.2\text{mm} \quad (i = 1,\ldots,5)$$

5.3.3 Materials and processing conditions

Three materials of contrasting shear thinning behaviour are used in the optimization:

- **Akulon F130-C PA6 (DSM)** is the least shear thinning of the three materials with $n = 0.85$ at a shear rate of 100s^{-1} at the extrusion temperature of 260°C.
- **LLDPE Montecatini A5 (Himont)** has $n = 0.50$ at a shear rate of 100s^{-1} at the extrusion temperature of 227°C.
- **Appryl 3010GN5 PP (ATO)** is most shear thinning of the three materials, with $n = 0.32$ at a shear rate of 100s^{-1} at the extrusion temperature of 250°C.

It should be noted that the power law (Equation 5.2) is not used to model the flow behaviour; instead a more flexible logarithmic polynomial fitted to the flow curves supplied by the manufacturers is used. The power law indices are included as an indication of the degree of shear thinning.

For the deterministic optimization (Section 5.4), two flow rates: 250kg/h and 80kg/h are studied for each material. For the robust design optimization (Sections 5.5 and 5.6), Montecatini A5 is studied at 80kg/h.

5.3.4 Choker bar profiles: bending or machining?

The usual procedure when using choker bar flow restrictors is to bend the bar in-situ to the required shape. The purpose of this section is to determine whether it is possible to bend the choker bar, from an intial flat profile, to the optimum shape before the choker bar mechanically fails; or whether it is necessary to machine the choker bar to the optimum shape.

The optimization of choker bar profiles described in this chapter is not constrained by any limitations on curvature such as would apply when the profile is to be achieved by the usual procedure of bending the bar. To establish if the required profile can be achieved by bending, a three-dimensional linear elastic analysis of displacing the bar to the optimized profile (see the bottom-left profile in Figure 5.10 for an illustration of the choker bar profile) for Appryl at high flow rate (most extreme case) is carried out. The cross section of the bar used in the modelled die is shown in Figure 5.9 with an indication of the FE mesh and boundary conditions applied in the analysis. The bar is free to slide in the Y′-direction but is prevented from moving in the X′-direction. The optimization results (Section 5.4) specify the vertical displacements, D, of the lower surface of the bar, which are achieved in practice by moving the bar a distance $\sqrt{2}D$ in the Y′-direction since the Y′ axis is at 45° to the horizontal. These displacements were applied to the choker bar model, which was meshed with Altair Hypermesh [5.20] using 98,033 tetrahedral elements and analyzed with Altair Optistruct [5.20]. The material properties are as for tool steel with a Young's modulus of E = 210,000N/mm^2 and a Poisson's ratio of $\nu = 0.3$.

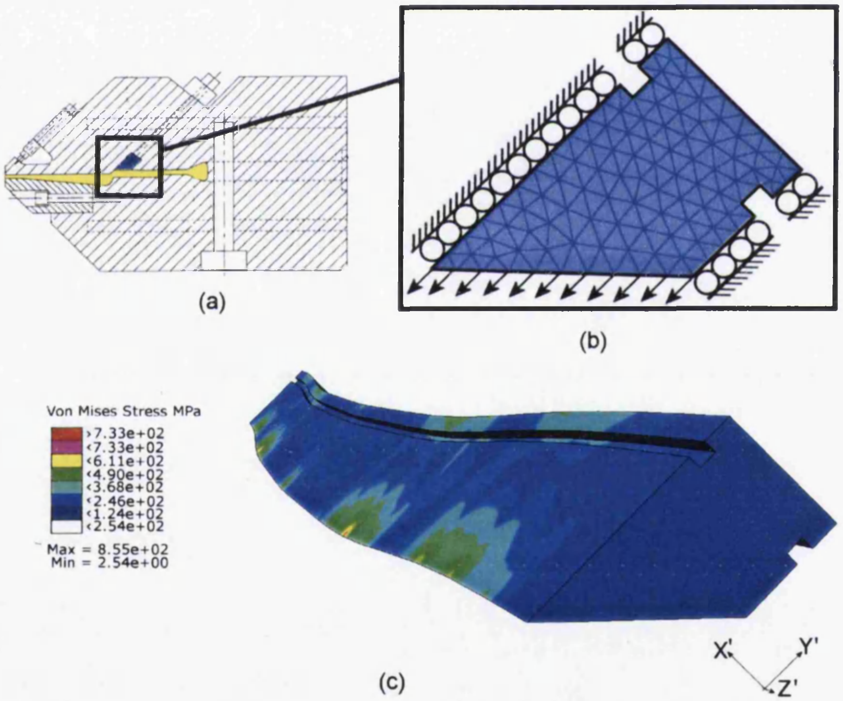


Figure 5.9: Modelling of the choker bar for stress analysis. (a) a cross section of the 3-d FE mesh. (b) an indication of the boundary conditions applied. (c) results of a linear stress analysis for bending of the bar to achieve the optimized profile computed for Appryl at 250 kg/h

The resulting stress field in the bar is shown in Figure 5.9(c). The maximum stress is 855Mpa, which is above the yield point for typical tool steel, leading to the conclusion that the optimized profile cannot be achieved without permanent deformation of the bar. In conclusion, it will be necessary, in some cases at least, to machine a restrictor bar to the computed optimum profile, with the possibility of minor adjustments on-line by bending. This, in turn, suggests the use of interchangeable restrictors, optimized and machined for specific applications, and providing the means to extend the range of operating conditions and materials for which a die can produce the required uniform flow distribution.

5.4 DETERMINISTIC OPTIMIZATION

For the materials and processing conditions given in Section 5.3.3, optimization is used to determine the optimum choker bar profiles that give the best possible exit flow uniformity from the slit die. The optimization is done assuming **ideal** conditions meaning that no noise is modelled in the system; therefore, the optimization can be considered as deterministic. Three optimization methods are used: GA, MARS and DOT (see Chapter 2 for further details on the methods) and their performances are compared with respect to efficiency and ability to obtain a global optimum.

5.4.1 COMPUTER IMPLEMENTATION

The general descriptions of DOT, GA and MARS are described in detail in Section 2.3. Here, the 'simulator' in the optimization procedure shown in Figure 1.3 is POLYSIM (described in Section 5.2). POLYSIM is used to calculate the performance of the designs specified by the optimizer, the optimizer returns a modified design vector $\mathbf{d}$, the new design variables are implemented and POLYSIM is re-run.

5.4.2 DOT - SETUP

The convergence criteria is as follows:

- the maximum absolute change in the objective function is less than 10^{-5} times the objective function value for the original design.
- the absolute value of the fractional change in the objective function from one iteration to the next is less than 10^{-4}.

Either of the conditions must be met for six consecutive iterations.

5.4.3 GA – SETUP

The GA was set up with a population size of 20, one-point crossover and one-point mutation, both with 100% probability and a combination of elitist and tournament selection. The chromosome length is 7 bits giving a design variable resolution of 0.024mm.

5.4.4 MARS – SETUP

The multiplicative response surface model is used with original sub-domain sizes of 0.15 for the Akulon, 0.25 for the Montecatini and 1.0 for the Appryl, a minimum sub-region size of 0.01, six plan points and re-use of previous sub-region results.

5.4.5 RESULTS AND DISCUSSION

Figures 5.10 to 5.14 show the deterministic optimization results. The initial design variable vector corresponds to the restrictor bar in the fully open position. Figure 5.10 shows the optimized restrictor profiles for the six cases.

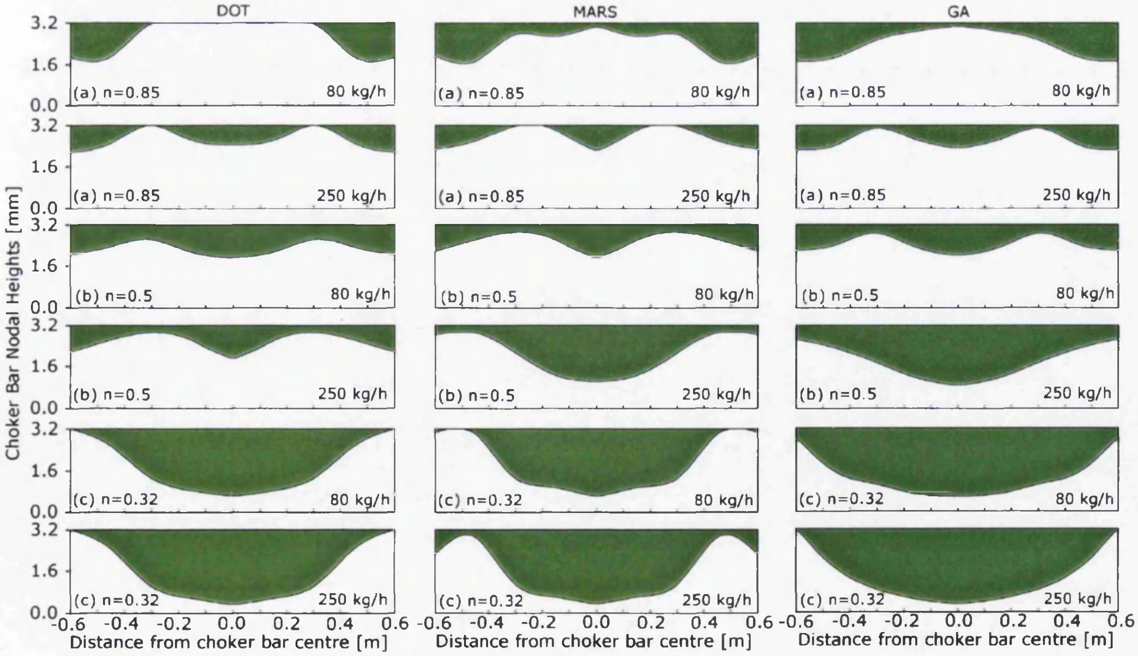


Figure 5.10: DOT(left), MARS(centre), GA(right) optimized choker bar profiles (green) for (a) Akulon, (b) Montecatini and (c) Appryl at the indicated flow rates

Figures 5.11-5.13 compare the initial flow rate distributions (restrictor fully open) with the final distributions obtained with the optimized restrictor profiles.

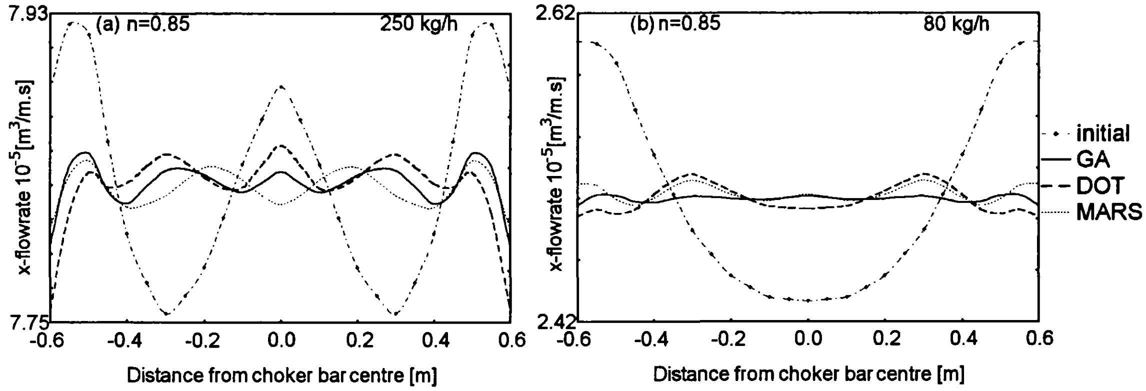


Figure 5.11: Exit flow distributions for Akulon at (a) 250kg/h and (b) 80kg/h flowrate. 'Initial' is a fully open choker bar and 'GA', 'DOT' and 'MARS' are for the respectively optimized restrictor profiles shown in Figure 5.10

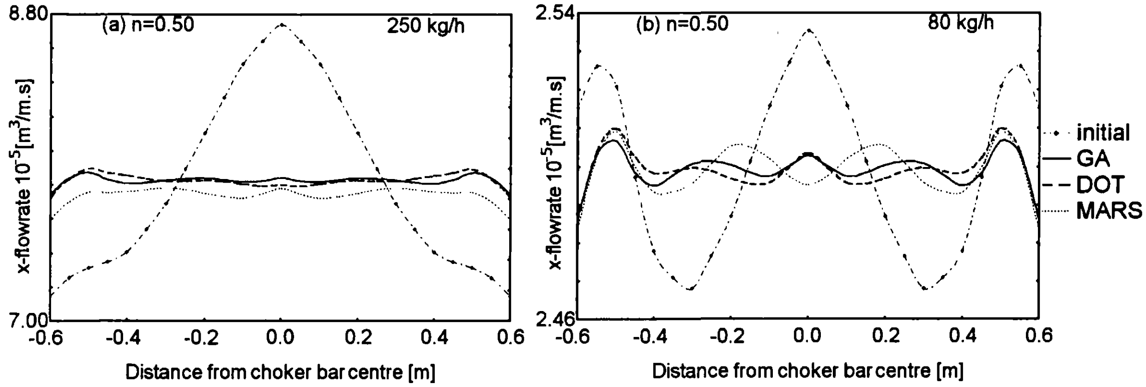


Figure 5.12: Exit flow distributions for Montecatini at (a) 250kg/h and (b) 80kg/h flowrate. 'Initial' is a fully open choker bar and 'GA', 'DOT' and 'MARS' are for the respectively optimized restrictor profiles shown in Figure 5.10

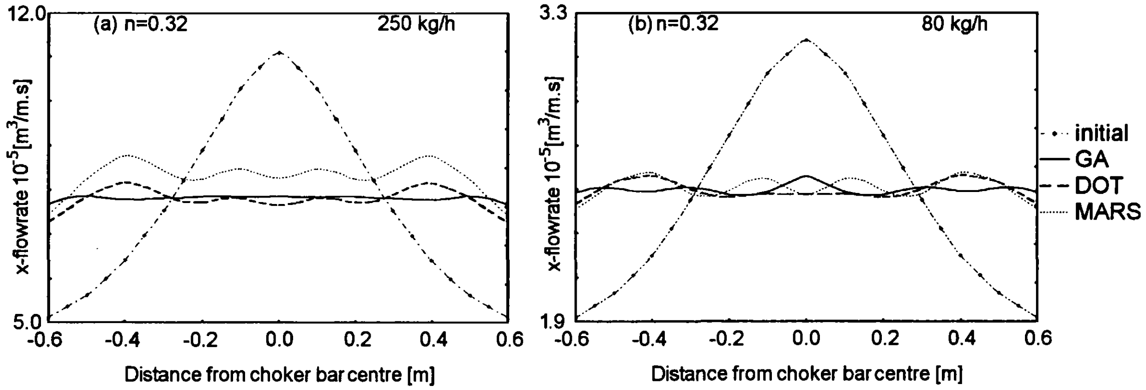


Figure 5.13: Exit flow distributions for Appryl at (a) 250kg/h and (b) 80kg/h flowrate. 'Initial' is a fully open choker bar and 'GA', 'DOT' and 'MARS' are for the respectively optimized restrictor profiles shown in Figure 5.10

Figure 5.14 tracks the convergence of the optimization procedures.

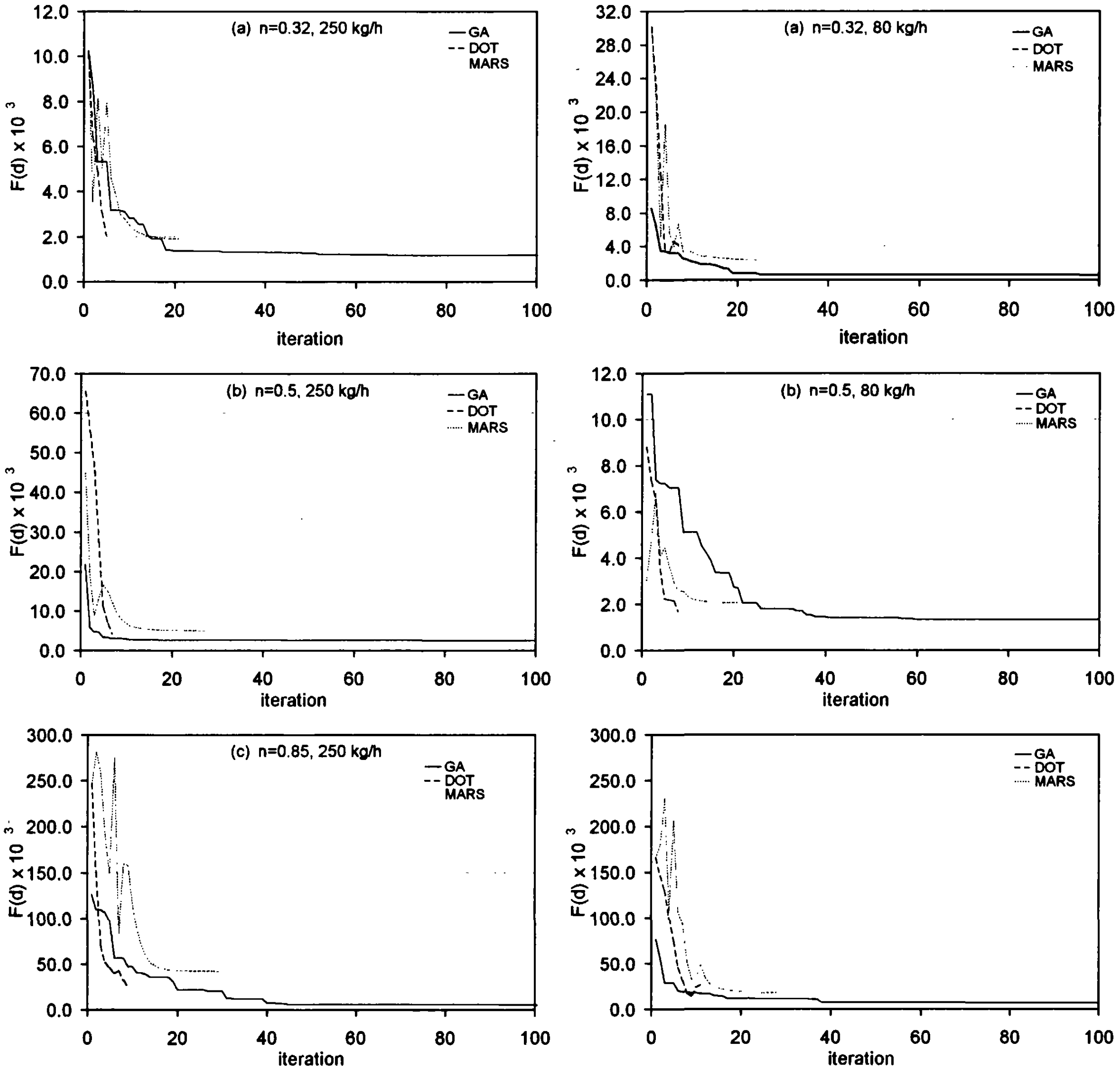


Figure 5.14: DOT, MARS and GA convergence history for (a) Akulon, (b) Montecatini and (c) Appryl

The reductions in the value of the objective function achieved by the optimization are summarized in Table 5.1. Overall, comparing the performances of the three methods used, the GA finds better solutions, but requires on average 10-times more function evaluations to find them than both MARS and DOT.

Table 5.2 summarizes the effectiveness of the optimization in terms of the maximum local deviations of the exit flow rate from the uniform (mean) value. Table 5.2 also shows the pressure drops through the die, the ratios between the values at high and low flow rates reflecting the increasing degrees of shear thinning from Akulon through to Appryl. From this Table, and from Figures 5.10 to 5.14 it can be seen that the die without restrictor correction operates better for the less shear thinning Akulon. At the lower operating throughput the initial flow rate distribution of this material (Figure 5.11(b)) shows excess flow at the die edges, which is partially corrected by die body deflection at the higher throughput (Figure 5.11(a)). For both, high and low throughputs, flow uniformity deviations are reduced to fractions of a percent by optimal adjustment of the restrictor, which is comfortably within the move limits of 0.1 to 3.2mm. For the slightly more shear thinning Montecatini the initial, uncorrected flow distribution at the lower flow rate (Figure 5.12(b)) shows a w-shaped profile, with high values now appearing at the centre as well as at the edges. At the higher operating throughput (Figure 5.12(a)), more material is channelled towards the centre as a result of die deflection, giving deviations of +12% and -8% about the mean. Again, the optimized restrictor profile is able to reduce deviations to within about a percent, without having to fully exploit the move limits. For the highly shear thinning Appryl, excess flow is found at the centre of the die, which is further exaggerated at the higher operating throughput (Figure 5.13). The initial deviations of 20 to 40% are reduced to a few percent by the optimized restrictor profile, which now more fully spans the move limits. It is apparent that in this particular extrusion application the restrictor adjustment is close to the limits of its effectiveness.

Case	PA6 Akulon F130-C *n*= 0.85		LLDPE Montecatini A5 *n*= 0.50		PP Appryl 3010 GN5 *n*= 0.32	
Throughput [kg/h]	80	250	80	250	80	250
Initial	30.1	10.1	8.8	65.5	165.7	246.6
Final – GA	0.6	1.2	1.3	2.5	7.0	4.3
Final – MARS	2.3	1.9	2.1	4.9	17.9	42.0
Final – DOT	3.4	2.1	1.7	4.1	13.7	24.7
Number of function evaluations (GA)	2000	2000	2000	2000	2000	2000
Number of function evaluations (MARS)	151	121	127	169	163	175
Number of function evaluations (DOT)	59	43	67	59	91	75

Table 5.1: Initial and final objective function values, F(d)×10^3, and number of function evaluations required to reach convergence

Case	PA6 Akulon F130-C $n = 0.85$		LLDPE MontecatiniA5 $n = 0.50$		PP Appryl 3010 GN5 $n = 0.32$	
Throughput [kg/h]	80	250	80	250	80	250
Initial local flow rate deviation [%]	+4.9	+1.8	+1.4	+12.0	+28.6	+43.9
	-3.5	-1.2	-1.3	-8.4	-22.7	-33.4
Final GA local flow rate deviation [%]	0.1	0.2	0.3	0.6	2.4	0.6
	-0.2	-0.5	-0.5	-1.0	-1.0	-1.7
Final MARS local flow rate deviation [%]	0.4	0.3	0.4	0.6	3.1	6.9
	-0.4	-0.5	-0.6	-2.1	-4.5	-12.5
Final DOT local flow rate deviation [%]	+0.6	+0.3	+0.4	+0.9	+2.6	+4.7
	-0.4	-0.9	-0.5	-1.3	-2.5	-6.6

Table 5.2: Maximum local exit flow rate variations expressed as a percentage about the mean, initial and final optimized values. The throughput flow rate and the representative power law indices are also shown

5.5 Robust design optimization – using a sample size of 10

RDO is applied to determine the choker bar profiles that give optimum melt flow distribution under **real** conditions for LLDPE Montecatini A5 (Himont) at a input flow rate of 80kg/h (Section 5.3 gives more information on the material). The optimization is carried out using DOT (see Chapter 2 for more information). In this section, RDO is done using a **uniformly** distributed sample size of 10 to estimate the mean and standard deviation at each point specified by the optimizer. Concentrating on the ability and the speed of generating a uniformly distributed Pareto set for the bi-objective problem, this section compares the performance of the PP method, the normal boundary intersection method (NBI) [4] and a method introduced here called the 'Pareto front marching method' using PP (PFM-PP). Note that this problem is referred to as the '10-sample' problem.

5.5.1 Problem formulation

As stated earlier in Chapter 4, to set up the RDO problem requires knowledge of the main sources of noise in the system. Based upon Pittman [5.21], it is assumed that the most important sources of noise and their likely variation are as listed in Table 5.3 meaning that there are eight noise factors. The real conditions are modelled by imposing variations of these key system parameters resulting in the RDO model in Figure 5.15.

(1)	Throughput flow rate of the polymer Q, ± 5% from nominal.
(2)	Operating temperature T, ± 10°C from nominal.
(3)	Raw material batch-to-batch variation, ± 0.1 from nominal power law index.
(4)-(8)	Variation in the die manufacture, ± 0.1mm from the nominal five shape variables.

Table 5.3: Parameters and their variations used in the slit die design

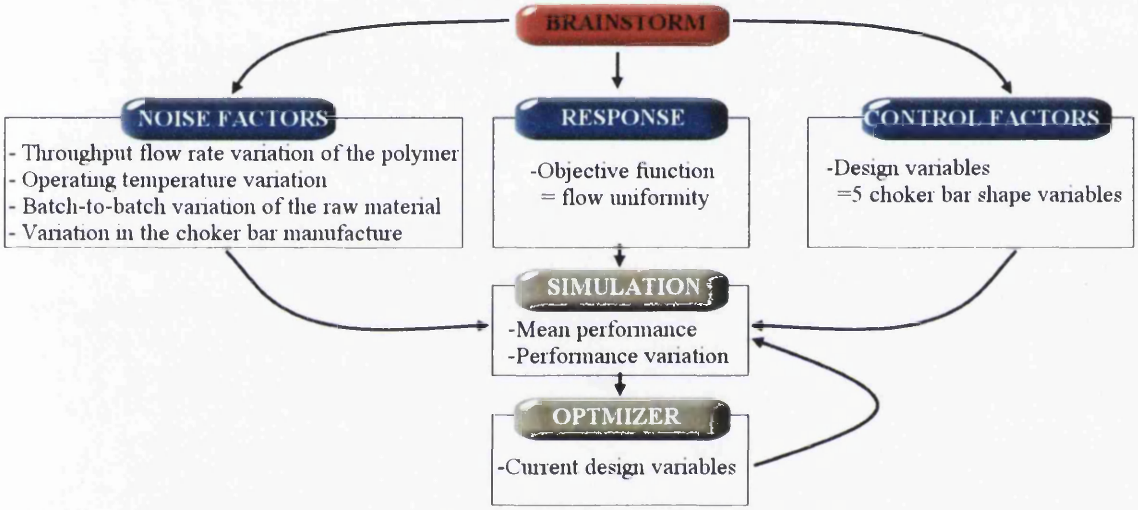

BRAINSTORM
NOISE FACTORS
- Throughput flow rate variation of the polymer
- Operating temperature variation
- Batch-to-batch variation of the raw material
- Variation in the choker bar manufacture
RESPONSE
-Objective function
= flow uniformity
CONTROL FACTORS
-Design variables
=5 choker bar shape variables
SIMULATION
-Mean performance
-Performance variation
OPTMIZER
-Current design variables

Figure 5.15: Robust design formulation for the slit die problem

5.5.2 Mean and Standard Deviation of the Performance

As stated earlier in Chapter 4, the RDO problem is bi-objective (mean and standard deviation), and a Pareto set of solutions is required to solve it. In order to determine the procedure for the RDO, a simplified model is used first. To generate the data for evaluating the mean and the standard deviation of the performance, 10 samples (10 simulations on 10 parallel processors) are run at each point specified by the optimizer. Each sample represents a slight variation of the noise factors from the nominal state. Only 10 simulations are run in order to reduce computational effort: the first is at the nominal state, the remaining nine (number of noise factors plus one) are generated using the AELH DoE (discussed in Chapter 3) for nine points and eight variables given in

Table 5.4. It should be noted that for this section a uniform distribution is applied to the parameter variations, meaning it is assumed that all values, within the ranges given in Table 5.3, are equally likely to occur in reality. Example 5.1 below is an example of how the DoE in Table 5.4 is used during a typical RDO cycle.

Parameter	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9
Δn	1	2	3	4	5	6	7	8	9
Q	7	2	6	8	5	4	1	3	9
T	9	3	2	5	4	6	1	8	7
d_1	3	7	2	5	9	8	4	6	1
d_2	4	6	1	9	3	7	8	5	2
d_3	6	5	7	3	1	9	8	2	4
d_4	4	6	1	5	7	3	8	9	2
d_5	8	9	5	1	2	7	4	6	3

Table 5.4: AELH DoE for nine points and eight variables, where each point is one simulation.

Example 5.1:

The nominal values for the parameters in Table 5.3 are:

- Δn = change in power law index
- flow rate = 3.00×10^{-3} m^3/m.s
- temperature = 227.0°C
- $\mathbf{d} = (d_1, d_2, d_3, d_4, d_5)$

The variation on these parameters is:

Parameter	Minimum	Maximum
Power law index, *n*	nominal − 0.1	nominal + 0.1
Flow, Q [m^3/m.s]	nominal × 0.95	nominal × 1.05
Temp, T [°C]	nominal − 10	nominal + 10
Shape variable, i [mm]	$d_i - 0.1$	$d_i + 0.1$

Table 5.5: Maximum and Minimum parameter variation from the nominal settings

The optimizer specifies the point (2.224, 2.590, 3.151, 2.639, 2.419) as the current set of shape variables. Using the DoE in Table 5.4 with 1 being equivalent to the minimum parameter value, 9 being equivalent to the maximum value, and intermediate values being interpolated between the two extremes (specified in Table 5.5), the DoE run matrix in Table 5.6 is formulated. These nine runs plus one run at the nominal state are run in parallel. Once all 10 runs are complete the mean and standard deviation of the performance (Equation 5.8), i.e. flow uniformity is calculated.

Parameter	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9
Δn	-0.100	-0.075	-0.050	-0.025	0.000	0.025	0.050	0.075	0.100
Q ($\times 10^{-3}$) [m^3/m.s]	3.075	2.888	3.038	3.113	3.000	2.963	2.850	2.925	3.150
T [°C]	237.0	222.0	219.5	227.0	224.5	229.5	217.0	234.5	232.0
d_1	2.174	2.274	2.149	2.224	2.324	2.299	2.199	2.249	2.124
d_2	2.565	2.615	2.490	2.690	2.540	2.640	2.665	2.590	2.515
d_3	3.144	3.126	3.163	3.088	3.051	3.200	3.181	3.070	3.107
d_4	2.614	2.664	2.539	2.639	2.689	2.589	2.714	2.739	2.564
d_5	2.494	2.519	2.419	2.319	2.344	2.469	2.394	2.444	2.369

Table 5.6: Example AELH DoE matrix for the slit die problem

5.5.3 COMPUTER IMPLEMENTATION

The implementation of the RDO procedure applied to the slit die problem is described in Appendix A2. Here, the optimizer used is DOT and, as in the deterministic optimization, the 'simulator' in the optimization procedure is POLYSIM (Section 5.2).

5.5.4 SOLUTION METHOD

Three multi-objective optimization methods are used to solve the RDO problem: PP, NBI and PFM-PP (see Chapter 3 for more details). Once the mean and standard deviation are calculated, the values are transformed according to which of the three methods is used and the optimizer uses the resulting performance information appropriately. The performances of PP, NBI and PFM-PP are compared, concentrating on the ability and the speed of generating a uniformly distributed Pareto set for the bi-objective RDO problem.

To define a practical range for generating the Pareto front, the two objectives are minimized separately to define the two utopia points. The first utopia point, f_1^*, is generated by minimizing the mean, the second, f_2^*, is generated by minimizing the standard deviation for a target mean of 7.0 (the approximate point where solutions become unacceptable). The two objectives have been normalized for easy result comparison and are in the form (mean, standard deviation). Normalizing the utopia points with the maximum values (7.006 and 2.245) gives the normalized utopia points $f_1^* = (0.5248, 1.0000)$ and $f_2^* = (1.0000, 0.4454)$. These two points define the two extreme Pareto solutions, between which the Pareto front is to be determined. In this study, 30 points along the Pareto front are generated using the NBI, PP (starting from a fully open choker bar) and PFM-PP methods.

5.5.5 Results and Discussion

Figure 5.16 gives the Pareto curves generated by (a) NBI, (b) PP and (c) PFM-PP. It can be seen that PFM-PP produced a smoother Pareto front than NBI and PP, and from Table 5.7 PFM-PP requires five-times fewer function evaluations than NBI and PP to generate the solutions. NBI and PP appear to have converged to local optima at several points, primarily as a result of using a gradient-based optimizer. PFM-PP seems to overcome this shortcoming by using the previous optimum solution as a starting point.

Method	Total number of function evaluations	Average number of function evaluations per Pareto point
NBI	6421	214
PP	7444	248
PFM-PP	1334	44

Table 5.7: Total and average number of function evaluations for each method

NBI and PFM-PP generate points in the region of the deterministic optimum (NRO in Figure 5.16), showing that the deterministic solution is non-Pareto. The compromise between mean and standard deviation of performance can be seen in Figure 5.17. An inspection of the curve, moving from f_1^* to f_2^*, shows that the worst performance (or maximum primary objective function value) decreases until solution 5. The next best solution is 15 which has the approximately the same worst performance (maximum primary objective function value) as 5, but the value of its minimum performance is larger than that of 5. Overall, solution 5 can be considered as the 'robust design' (RD). The reason for this is that, although the solutions become more predictable, i.e. have less variation moving from f_1^* to f_2^*, they also show that performance gradually decreases.

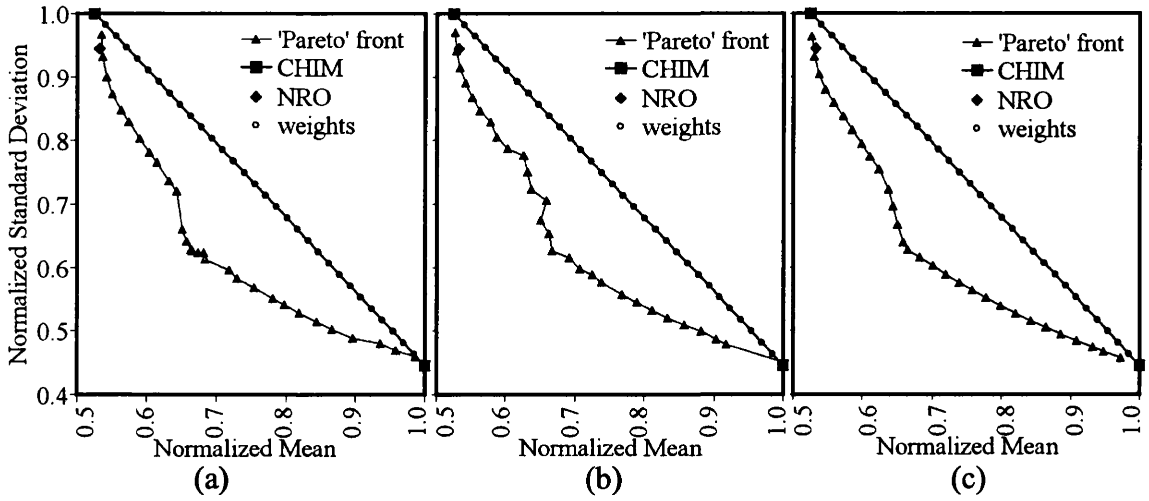


Figure 5.16: Results using a sample size of 10: (a) Pareto curve using NBI; (b) Pareto curve using PP; (c) Pareto curve using PFM-PP.

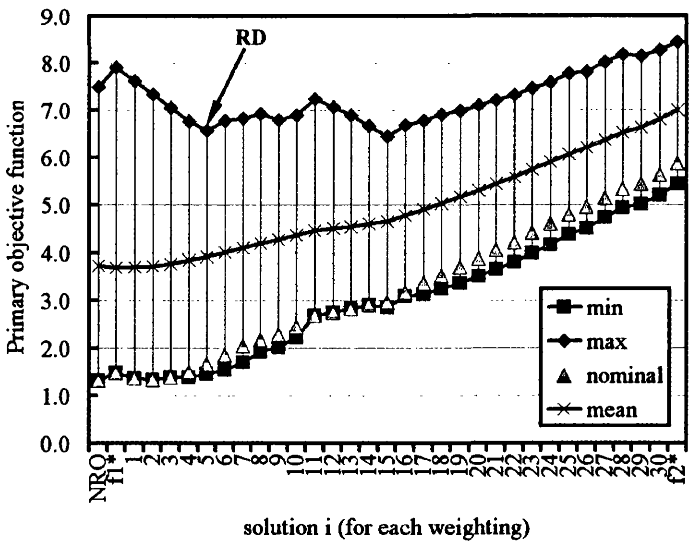


Figure 5.17: High-low plot of performance for solutions along the Pareto front found by PFM-PP for a sample size of 10

The shape of the RD choker bar is shown in Figure 5.18 and the corresponding best, worst (minimum and maximum of Equation 5.8 respectively) and nominal exit flow rates are depicted in Figure 5.19. The best and worst exit flow rate distributions correspond to run 6 and run 8 respectively in Table 5.4. It can be seen from Figure 5.19

that the average flow rate for the nominal case is higher than for the best and worst cases. This is expected, since the input flow rate for the nominal is greater than runs 6 and 8.

The differences in the design variables between the RD and the NRO can be seen in Table 5.8, where the maximum difference is about 0.05mm. Thus, only a very small change in the choker bar shape is required to move from the NRO to the RD.

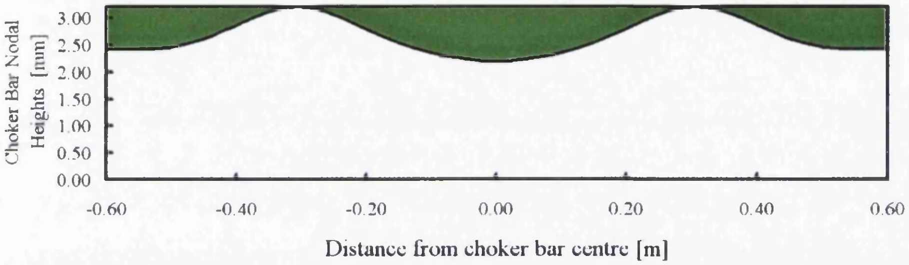


Figure 5.18: Choker bar profile for the RD, solution 5 in Figure 5.17

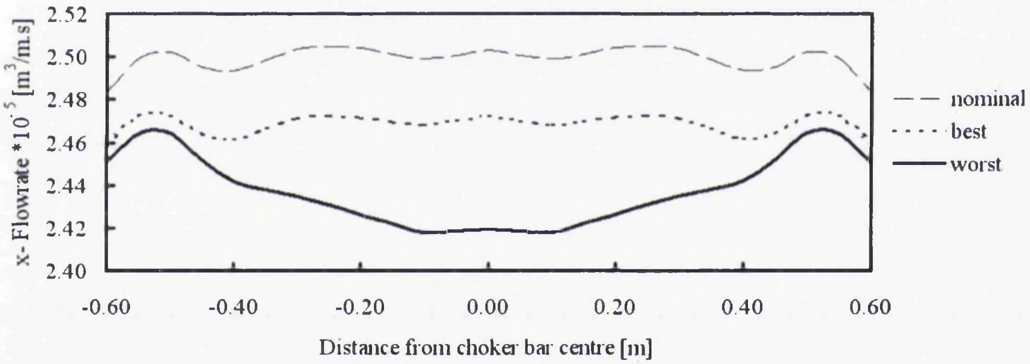


Figure 5.19: Exit flow rates for the RD, solution 5 in Figure 5.17

Solution	d_1	d_2	d_3	d_4	d_5
NRO	2.224	2.590	3.151	2.639	2.419
RD	2.211	2.618	3.200	2.585	2.424

Table 5.8: Design variable values in mm for the NRO and the RD

The overall progression in the choker bar profiles for the 10-sample problem, starting with solution 1 for weight 1 moving to solution 30 for weight 30, can be seen in Figure 5.20. It can be seen that up to weight 5 the choker bar maintains a similar shape. A slightly different shape develops between 6 and 10 and at weight 11 it can be seen that between -0.2m and -0.4m approximately (and similarly +0.2m and +0.4m), from the choker bar centre the move limits are reached. The width of these bands increase thereafter to weight 30 where the profile becomes virtually fully open except for a small region at the centre and the edges. The optimizer wants to move the variables further but because these regions are at the move limits, i.e. fully open, it cannot.

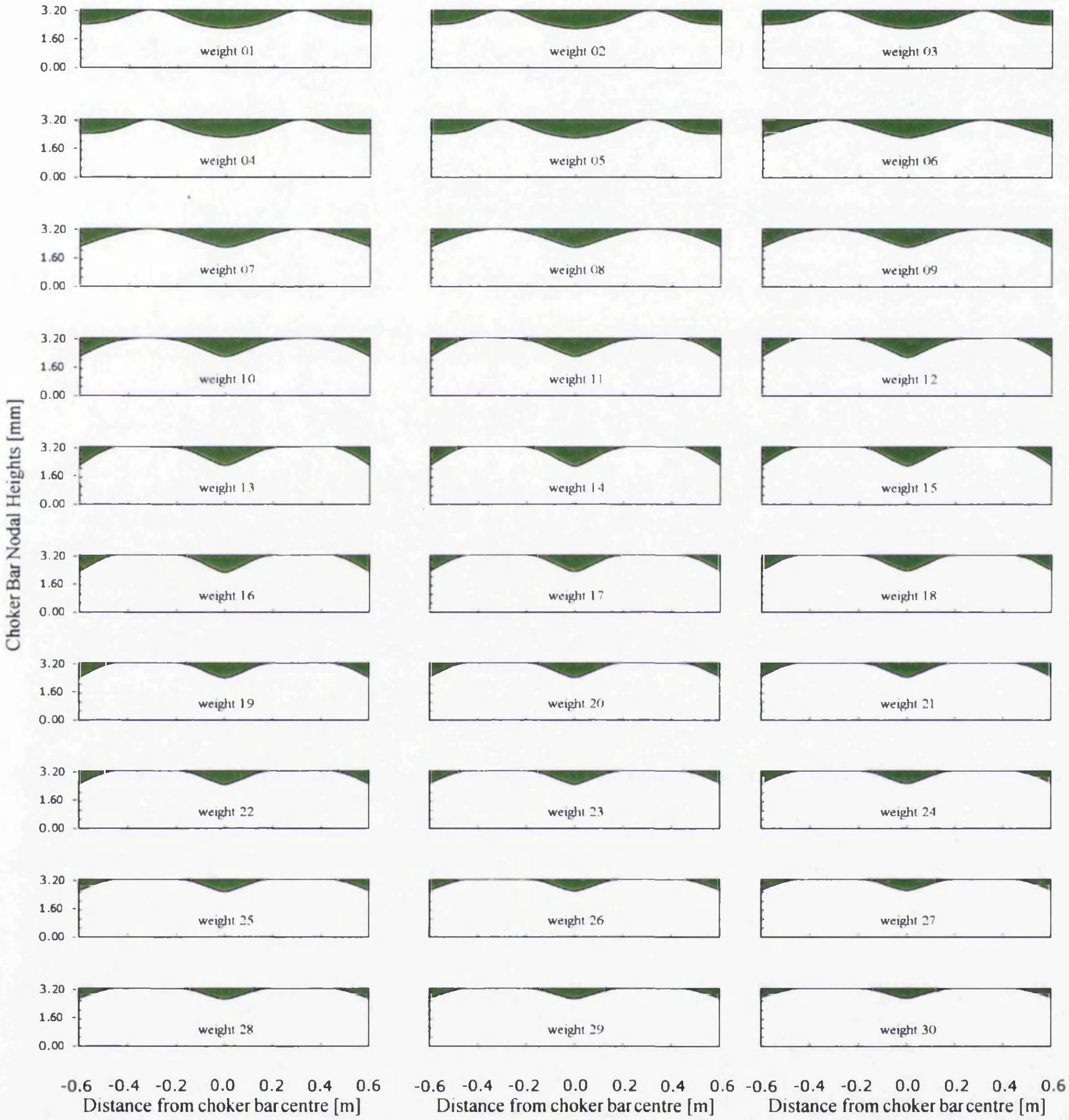
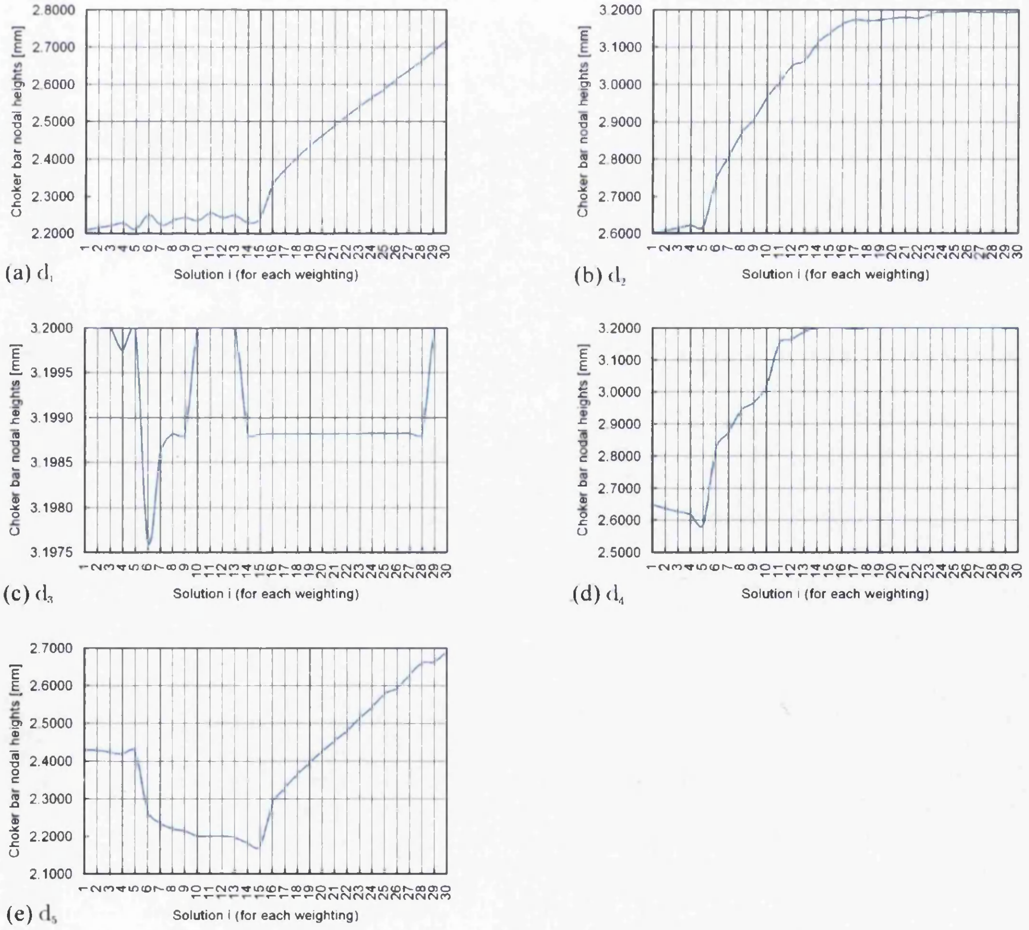


Figure 5.20: Choker bar profiles for the 30 Pareto solutions in Figure 5.16(c) found using a sample size of 10

The variation of each of the five design variables for each of the 30 Pareto solutions to the 10-sample problem can be seen in more detail in Figure 5.21 and Table 5.9 from which, the following observations can be made:

- All of the design variable values lie between 2.0mm and 3.2mm and the variations are about 0.6mm, except for d_3, which remains almost constant at the move limit of 3.2mm (fully open).
- d_1 is between 2.2 and 2.3mm up to solution 15 and then increases monotonically to about 2.7mm at solution 30.
- d_2 is about 2.6mm until solution 5, from solution 6 to 16 it increases (roughly) monotonically from 2.6mm to 3.15mm, thereafter the values increase to the move limit of 3.2mm (fully open).
- d_4 is between 2.58 and 2.65mm for the first five solutions and then increases to the move limit of 3.2mm (fully open) by solution 14, thereafter it remains at 3.2mm.
- d_5 is roughly constant at 2.42mm for the first five solutions. There is a step change at solution 6 to 2.25mm. From solution 6 to 15, d_5 decreases to 2.18mm and thereafter increases monotonically to 2.7mm at solution 30.



(a) d_1

(b) d_2

(c) d_3

(d) d_4

(e) d_5

Figure 5.21: Plots of each design variable value for each of the 30 Pareto solutions for a sample size of 10

Solution	d_1	d_2	d_3	d_4	d_5
NRO	2.224	2.590	3.151	2.639	2.419
f_1^*	2.198	2.591	3.200	2.656	2.427
1	2.209	2.601	3.200	2.649	2.428
2	2.215	2.607	3.200	2.636	2.427
3	2.221	2.613	3.200	2.626	2.422
4	2.228	2.621	3.200	2.617	2.417
5	2.211	2.618	3.200	2.585	2.424
6	2.251	2.745	3.198	2.823	2.266
7	2.222	2.807	3.199	2.873	2.234
8	2.235	2.868	3.199	2.942	2.219
9	2.242	2.907	3.199	2.967	2.213
10	2.235	2.965	3.200	3.020	2.198
11	2.255	3.007	3.200	3.152	2.198
12	2.241	3.049	3.200	3.164	2.200
13	2.248	3.064	3.200	3.188	2.195
14	2.228	3.112	3.199	3.200	2.181
15	2.240	3.139	3.199	3.200	2.171
16	2.329	3.163	3.199	3.200	2.285
17	2.370	3.174	3.199	3.197	2.328
18	2.403	3.171	3.199	3.200	2.364
19	2.434	3.174	3.199	3.200	2.394
20	2.461	3.178	3.199	3.200	2.424
21	2.488	3.180	3.199	3.200	2.453
22	2.516	3.178	3.199	3.200	2.479
23	2.539	3.189	3.199	3.200	2.513
24	2.565	3.194	3.199	3.200	2.542
25	2.586	3.194	3.199	3.200	2.578
26	2.614	3.195	3.199	3.200	2.594
27	2.638	3.192	3.199	3.200	2.629
28	2.661	3.194	3.199	3.200	2.660
29	2.689	3.192	3.200	3.199	2.664
30	2.718	3.194	3.200	3.199	2.690
f_2^*	2.745	3.198	3.200	3.200	2.727

Table 5.9: Design variable values for the NRO, the 30 Pareto solutions shown in Figure 5.16(c) and the utopia points f_1^* and f_2^*

5.6 Robust design optimization – using a sample size of 100

RDO is applied to determine the choker bar profiles that give optimum melt flow distribution under **real** conditions for LLDPE Montecatini A5 (Himont) at a input flow rate of 80kg/h (Section 5.3 gives more information on the material). The optimization is carried out using DOT (see Chapter 2 for more information). In this Section a more realistic, but more computationally intensive, **normally** distributed sample of size 100 is used in the same way as in Section 5.5, where a sample size of 10 was used. Here PFM-PP is used to solve the bi-objective problem. The differences between the results of the two sample sizes are then discussed. Note, the problem using a sample size of 10 will be written as '10-sample', and the problem using a sample size of 100 as '100-sample'.

5.6.1 PROBLEM FORMULATION

The problem is formulated as in Section 5.5.1

5.6.2 MEAN AND STANDARD DEVIATION OF THE PERFORMANCE

To generate the data for evaluating the mean and the standard deviation of the performance, 100 samples are run at each point specified by the optimizer. Each sample represents a slight variation of the noise factors from the nominal state. The first is at the nominal state, the remaining 99 are generated using the normally distributed AELH DoE (discussed in Chapter 3) for 99 points and eight variables as given in Table A1.5. It should be noted that in this section a normal distribution is applied to the parameter variations, meaning it is assumed that all values, within the ranges given in Table 5.3, are normally distributed about the mean.

5.6.3 COMPUTER IMPLEMENTATION

The implementation of the RDO procedure applied to the slit die problem is described in Appendix A2. Here, the optimizer used is DOT and as in the deterministic optimization, the 'simulator' in the optimization procedure is POLYSIM (Section 5.2).

5.6.4 SOLUTION METHOD

From Section 5.5 it can be seen that, for the current implementation, PFM-PP requires fewer iterations and produces a more uniform Pareto front than PP and NBI. Hence PFM-PP is used here to solve the RDO problem (see Chapter 3 for more details). Again, as in Section 5.5, a practical range for generating the Pareto front is required. The two objectives are minimized separately to define the two utopia points. The first utopia point f_1^*, is generated by minimizing the mean, the second f_2^*, by minimizing the standard deviation for a target mean of 7.0 (the approximate point where solutions become unacceptable). Normalizing the utopia points with the maximum values of the mean and standard deviation (7.009 and 1.039) gives the normalized utopia points $f_1^* = (0.3460, 1.0000)$ and $f_2^* = (1.0000, 0.4562)$. These two points define the two extreme Pareto solutions, between which the Pareto front is to be determined. In this study, 30 points along the Pareto front are generated.

5.6.5 Results and Discussion

Comparison between the two curves generated for the 10-sample and 100-sample problems can be seen in Figure 5.22. The '10 samples' curve was generated using the 30 solutions generated in Section 5.5 (listed in Table 5.9 (weights 1 to 30)) and running the same normally distributed DoE (Table A1.5) used for the 100-sample RDO. The following can be seen from Figure 5.22: (i) the 'Pareto' solutions for the 10-sample problem do not lie on the Pareto front for the 100-sample problem; (ii) the deterministic solution, NRO, lies very close to the 'Pareto' front, indicating that it is a possible robust design; (iii) the PFM-PP method has generated a uniform spread of points.

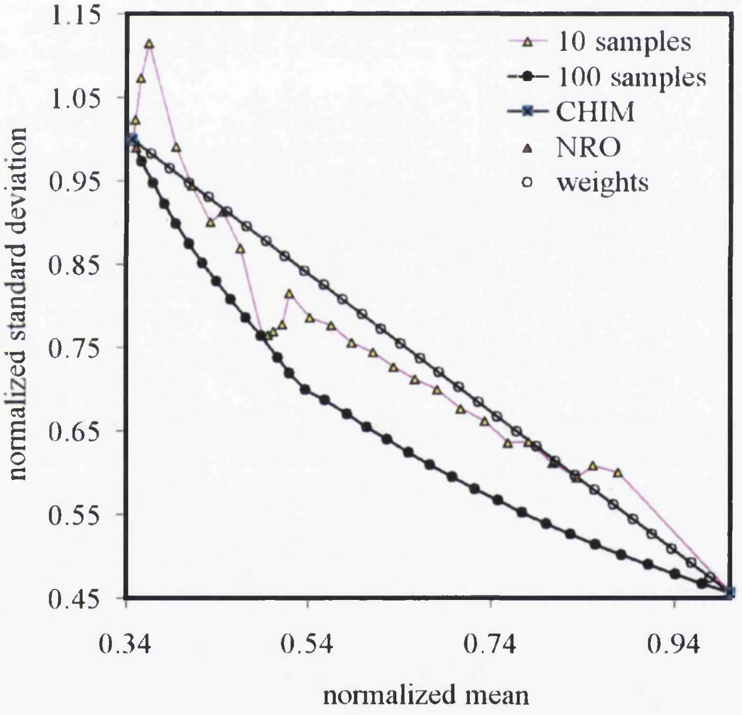
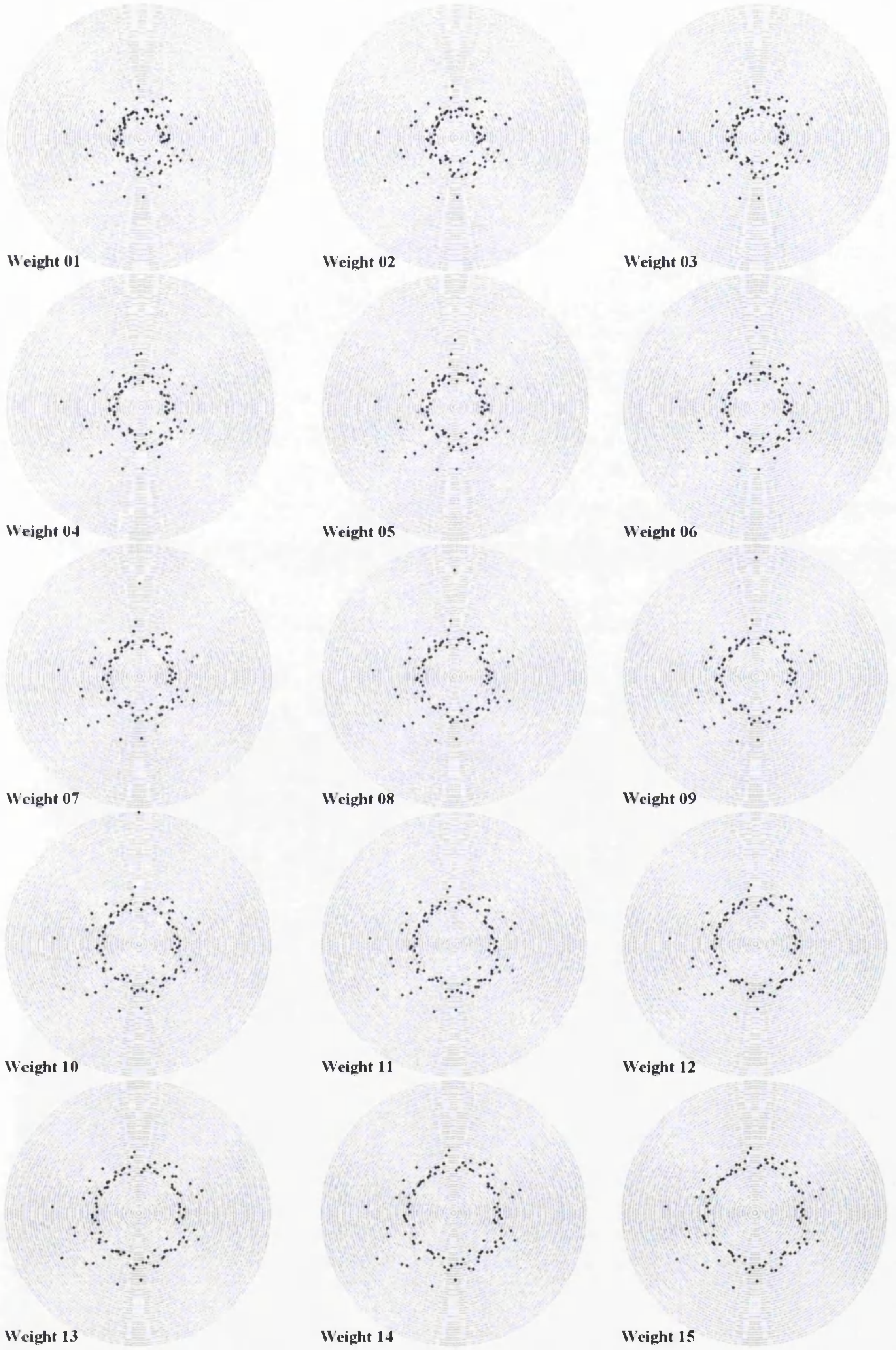


Figure 5.22: Results: Pareto curve for the 100-sample problem compared to the curve generated using the 10-sample problem results, both using a normally distributed sample

The actual performance variation can be seen in Figures 5.23 and 5.24, which contain the same basic data. Figure 5.23 shows the raw data in a plot of the performance of each sample for each of the 30 Pareto solutions and Figure 5.24 is an accumulation of the data into histograms. An explanation of Figures 5.23 and 5.24 is given in Section A2.3.



Weight 01

Weight 02

Weight 03

Weight 04

Weight 05

Weight 06

Weight 07

Weight 08

Weight 09

Weight 10

Weight 11

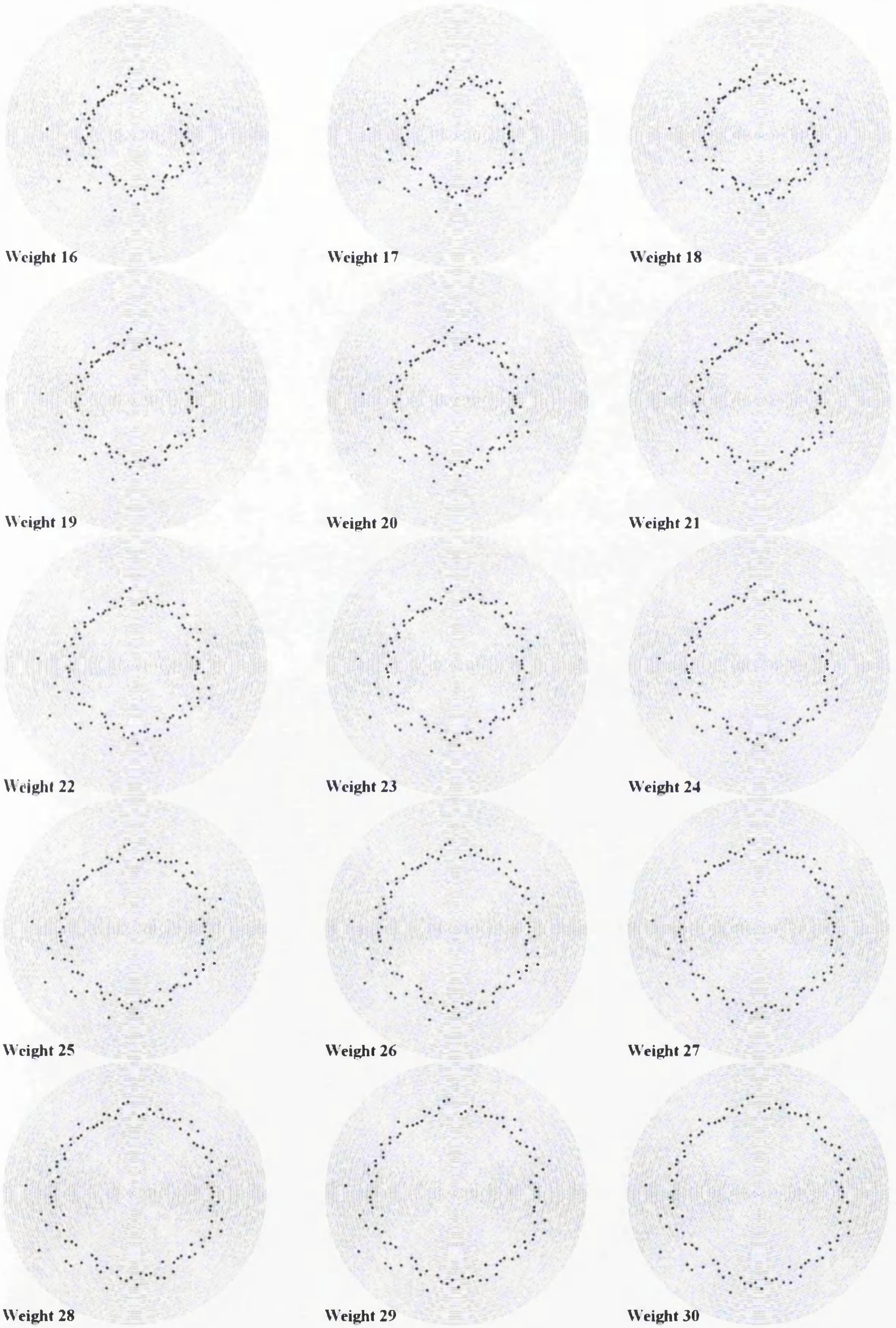
Weight 12

Weight 13

Weight 14

Weight 15

Figure 5.23: Performance scatter for each of the 30 Pareto solutions for the 100-sample problem shown in Figure 5.22. The centre of each plot represents a value of zero for the primary objective function (Equation 5.8) and the outer boundary is a value of 10, the radial co-ordinate is the solution number



Weight 16 Weight 17 Weight 18

Weight 19 Weight 20 Weight 21

Weight 22 Weight 23 Weight 24

Weight 25 Weight 26 Weight 27

Weight 28 Weight 29 Weight 30

Figure 5.23: ***(continued)***

It can be seen from Figure 5.23 that, moving from the solution for weight 1 to the solution for weight 30: (i) the overall performance of the samples decrease because the solutions move further from the centre of the plots where the ideal performance lies; (ii) the sample performance becomes more predictable. Both these observations are expected because as one moves from weight 1 to weight 30 the mean has increasingly less influence and the standard deviation has increasingly more influence on the objective. This behaviour can also be seen in Figure 5.24, which shows histograms of the performance for each of the Pareto solutions. Again, it can be seen that moving from weight 1 to weight 30 the spread of the solutions decreases and the mean increases.

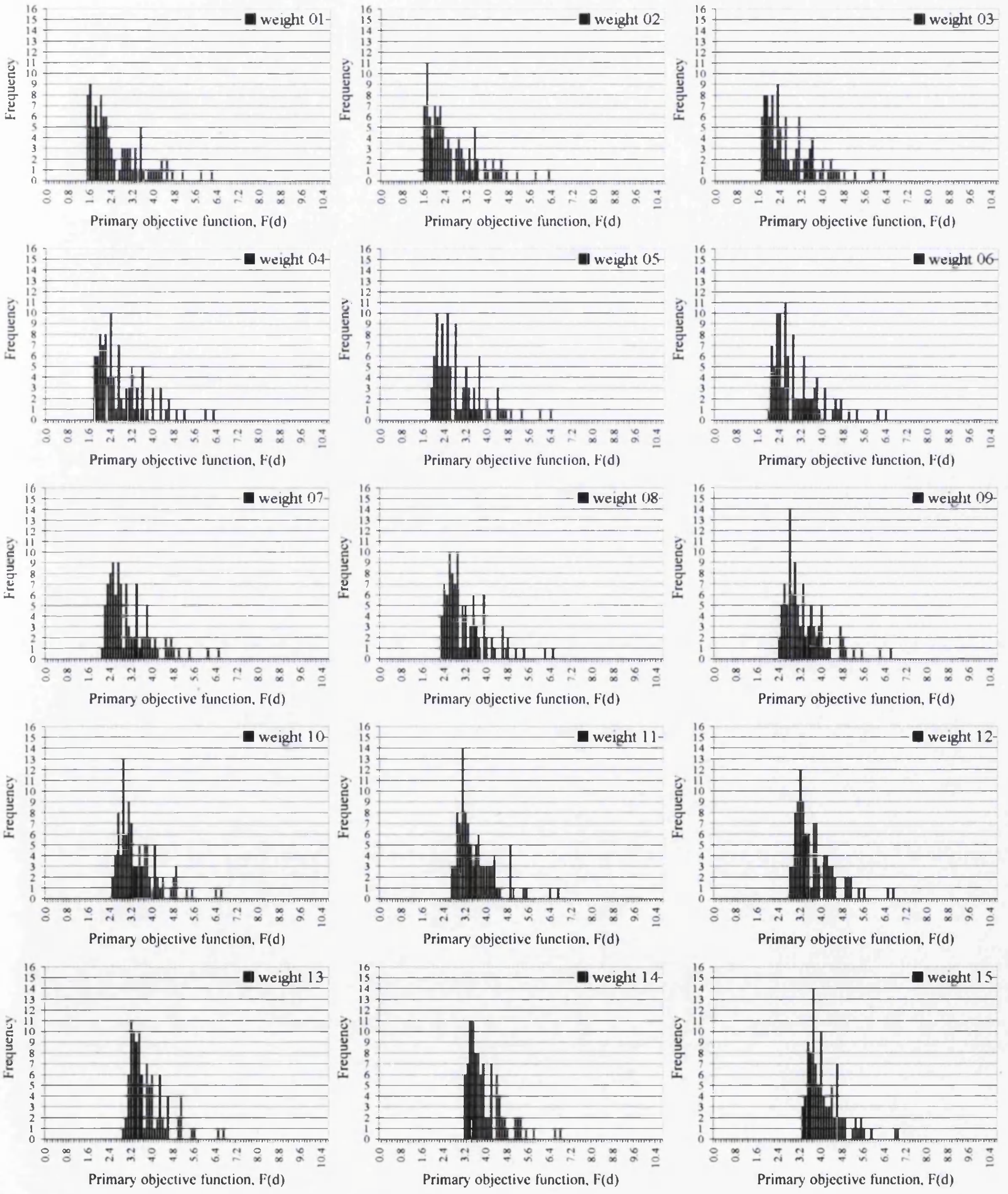


Figure 5.24: Histograms showing the variation of the performance of each of the 30 Pareto solutions for the 100-sample problem shown in Figure 5.22

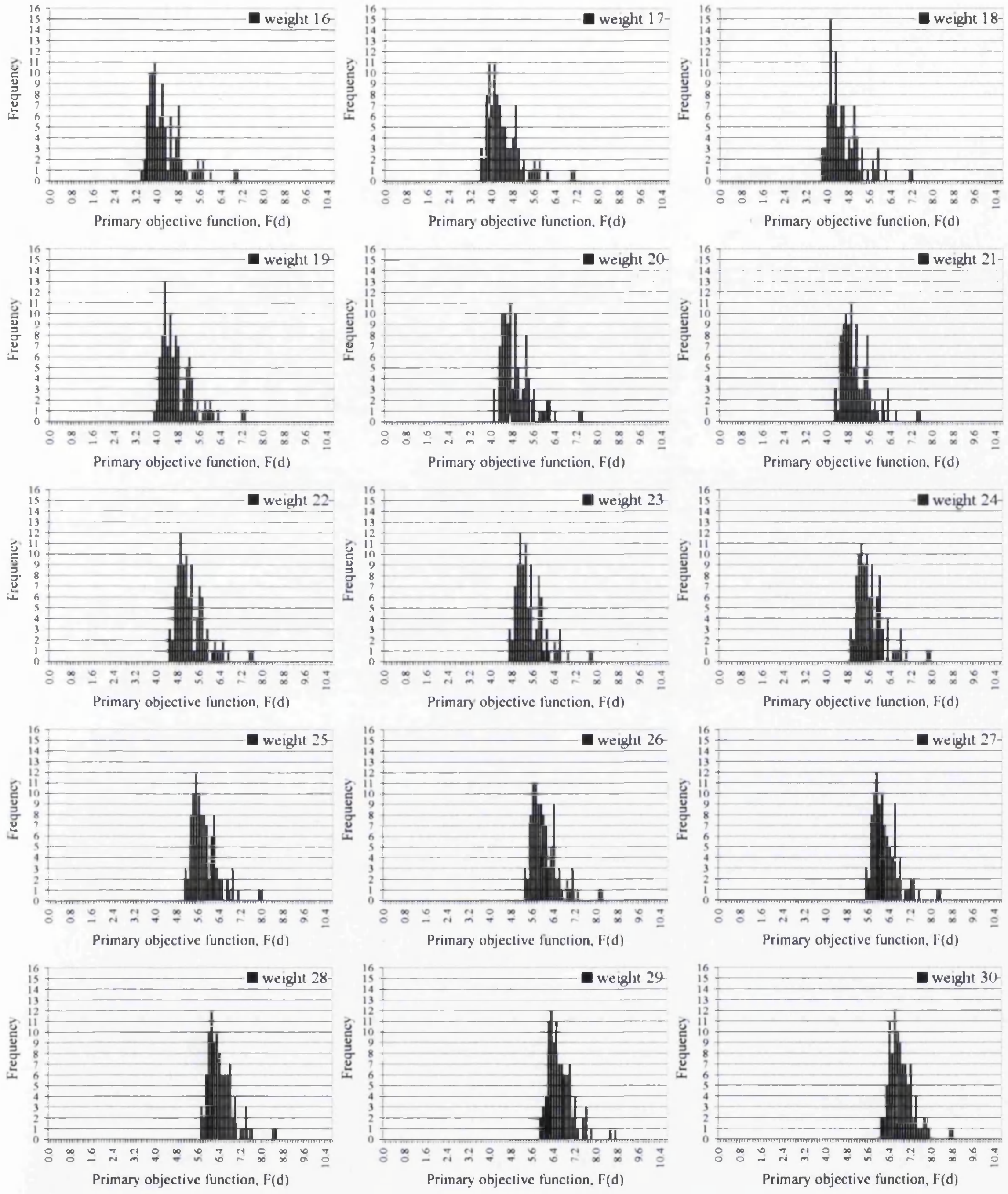


Figure 5.24: ***(continued)***

The overall progression in the choker bar profiles for the 100-sample problem, starting with solution 1 for weight 1 moving to solution 30 for weight 30, can be seen in Figure 5.25. It can be seen, that the profiles are slightly different to those for the 10-sample problem in Figure 5.20 and there is no obvious point at which there is a significant change in the profile, unlike for the 10-sample problem where the change from solution for weight 5 to the solution for weight 6 is clearly visible. Instead, the change in the profile is gradual from a 'wavy m' shape for the weight 1 solution to being almost fully open, except at the centre, for the weight 30 solution. This may account for the reason that there is a 'kink' in the 'Pareto' curve for the 10-sample problem and that the 'Pareto' curve for the 100-sample problem is smoother.

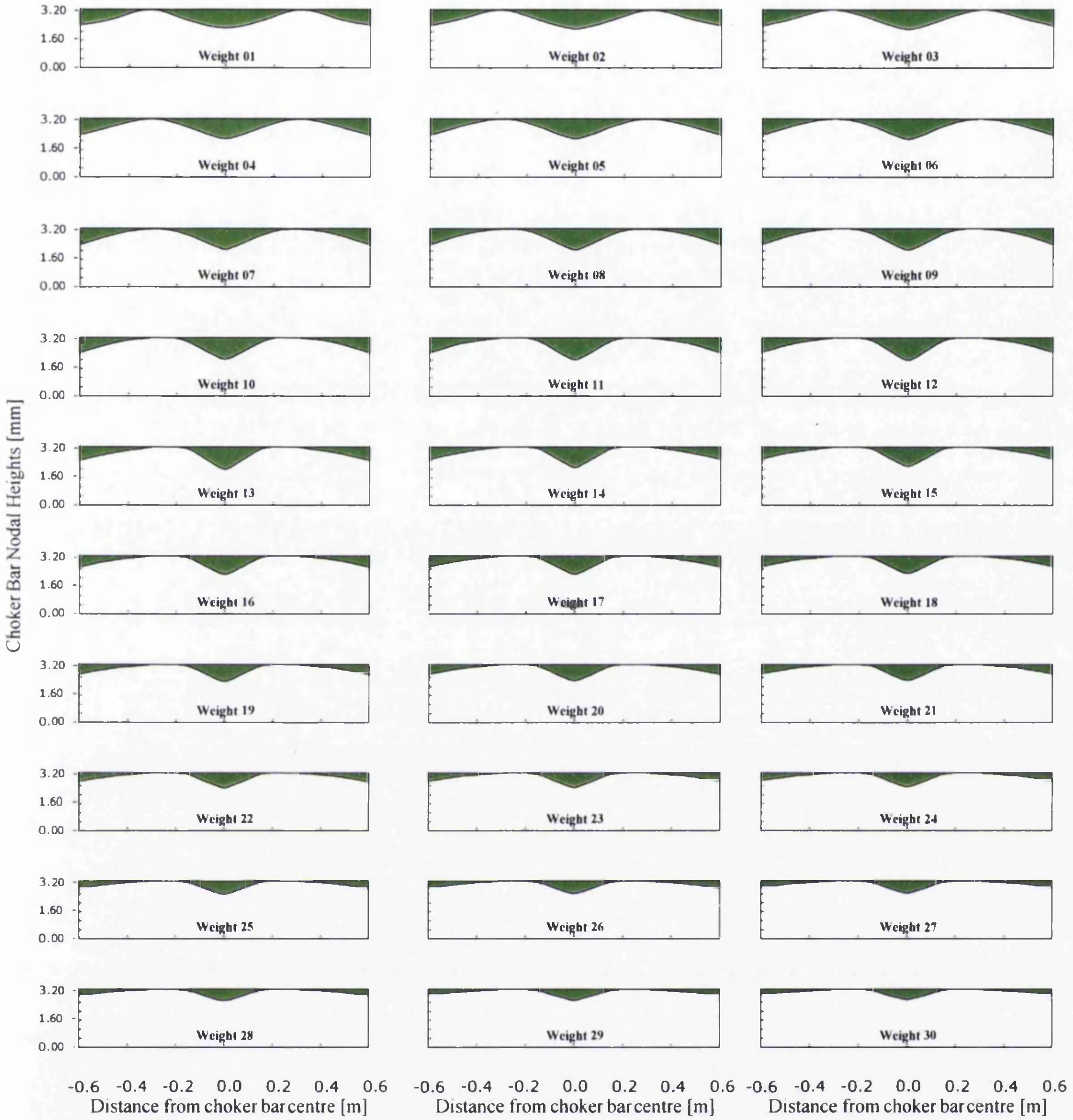
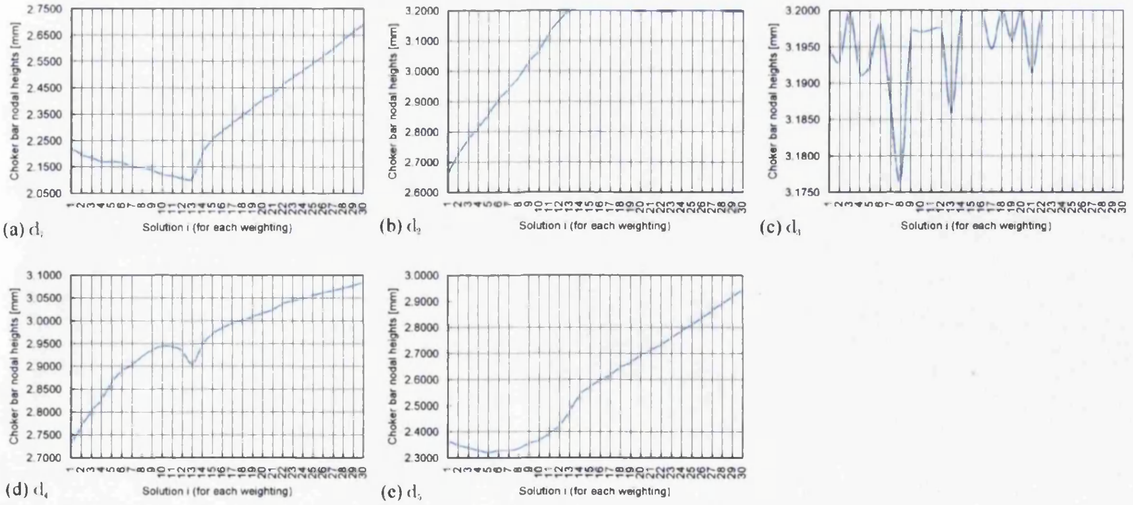


Figure 5.25: Choker bar profiles for the 30 Pareto solutions to the 100-sample problem in Figure 5.22

The variation of each of the five design variables for each of the 30 Pareto solutions to the 100-sample problem can be seen in more detail in Figure 5.26 and Table 5.10 from which, the following observations can be made:

- As for the 10-sample problem, all of the design variable values lie between 2.0mm and 3.2mm and the variations are roughly 0.6mm, except for d_3, which remains almost constant at the upper move limit of 3.2mm (fully open).

- For all of the variables except d_3, solution 13 is the point where the trend in the variation of the variables between one solution and the next changes.
- By solution 30 d_2, d_3 and d_4 are all almost at the move limits and only d_1, which affects the centre of the flow channel, and d_5, which affects the edge of the flow channel, have any influence on the flow distribution.
- d_1 decreases from 2.2mm to 2.1mm up to solution 13, thereafter increases (roughly) monotonically to about 2.7mm at solution 30. The overall variation in d_1 is approximately 0.6mm.
- d_2 increases monotonically from 2.66mm at solution 1 and reaches the move limit of 3.2mm at solution 13 and thereafter it remains at 3.2mm. The overall variation in d_2 is approximately 0.5mm.
- d_4 increases (roughly) linearly from 2.73mm at solution 1 to 2.95mm at solution 10. d_4 then decreases to 2.90mm at solution 13 and from solution 14 to solution 30 d_4 increases (roughly) linearly from 2.95mm to 3.08mm. The overall variation in d_4 is approximately 0.35mm.
- d_5 decreases from 2.36mm at solution 1 to 2.32mm at solution 5, from solution 6 to 14 d_5 increases exponentially to 2.54mm. Thereafter d_5 increases linearly to 2.95mm at solution 30. The overall change in d_5 is 0.6mm.



(a) d_1 (b) d_2 (c) d_3 (d) d_4 (e) d_5

Figure 5.26: Plots of each design variable value for each of the 30 Pareto solutions for the 100-sample problem

Solution	d_1	d_2	d_3	d_4	d_5
NRO	2.224	2.590	3.151	2.639	2.419
f_1^*	2.216	2.569	3.199	2.614	2.419
1	2.220	2.663	3.195	2.729	2.364
2	2.194	2.725	3.193	2.767	2.346
3	2.183	2.774	3.200	2.801	2.336
4	2.169	2.812	3.191	2.826	2.326
5	2.169	2.857	3.193	2.863	2.317
6	2.164	2.906	3.198	2.890	2.325
7	2.148	2.941	3.187	2.902	2.325
8	2.143	2.979	3.176	2.920	2.333
9	2.136	3.031	3.197	2.935	2.354
10	2.120	3.069	3.197	2.944	2.365
11	2.114	3.128	3.197	2.943	2.391
12	2.105	3.168	3.197	2.933	2.421
13	2.107	3.200	3.186	2.904	2.476
14	2.202	3.200	3.200	2.947	2.539
15	2.253	3.200	3.200	2.971	2.570
16	2.286	3.200	3.200	2.984	2.595
17	2.315	3.200	3.195	2.994	2.616
18	2.346	3.200	3.200	3.001	2.645
19	2.374	3.200	3.196	3.009	2.666
20	2.405	3.197	3.200	3.016	2.690
21	2.425	3.200	3.191	3.024	2.712
22	2.462	3.197	3.200	3.037	2.733
23	2.490	3.197	3.200	3.044	2.759
24	2.515	3.200	3.200	3.049	2.784
25	2.543	3.200	3.200	3.055	2.809
26	2.571	3.199	3.200	3.061	2.835
27	2.598	3.200	3.200	3.066	2.863
28	2.629	3.199	3.200	3.071	2.890
29	2.662	3.194	3.200	3.077	2.917
30	2.691	3.197	3.200	3.084	2.945
f_2^*	2.718	3.200	3.200	3.092	2.976

Table 5.10: Design variable values for the NRO, the 30 Pareto solutions shown in Figure 5.22 and the utopia points f_1^* and f_2^* for the 100-sample problem

The corresponding exit flow-rate distributions for NRO and several solutions on the 100-sample 'Pareto' curve in Figure 5.22 are shown in Figure 5.27. For clarity only the best, worst and nominal samples are shown, it should be noted that, as can be seen in Figures 5.23 and 5.28, the performance of the majority of the samples have a flow uniformity similar to the 'best' curve and not the 'worst' owing to the normal distribution of the scatter factors. It can be seen from Figure 5.27 that for NRO and the solution to weight 1, the difference in shape of the flow-rate distribution curves for the 'best' and 'worst' cases is much more pronounced than for weight 30. Also, the 'worst' flow curve for weight 1 is as good as the 'best' curve for weight 30, this is backed up in Figure 5.28 since the lowest value of the primary objective function for weight 30 is approximately the same as the highest for weight 1.

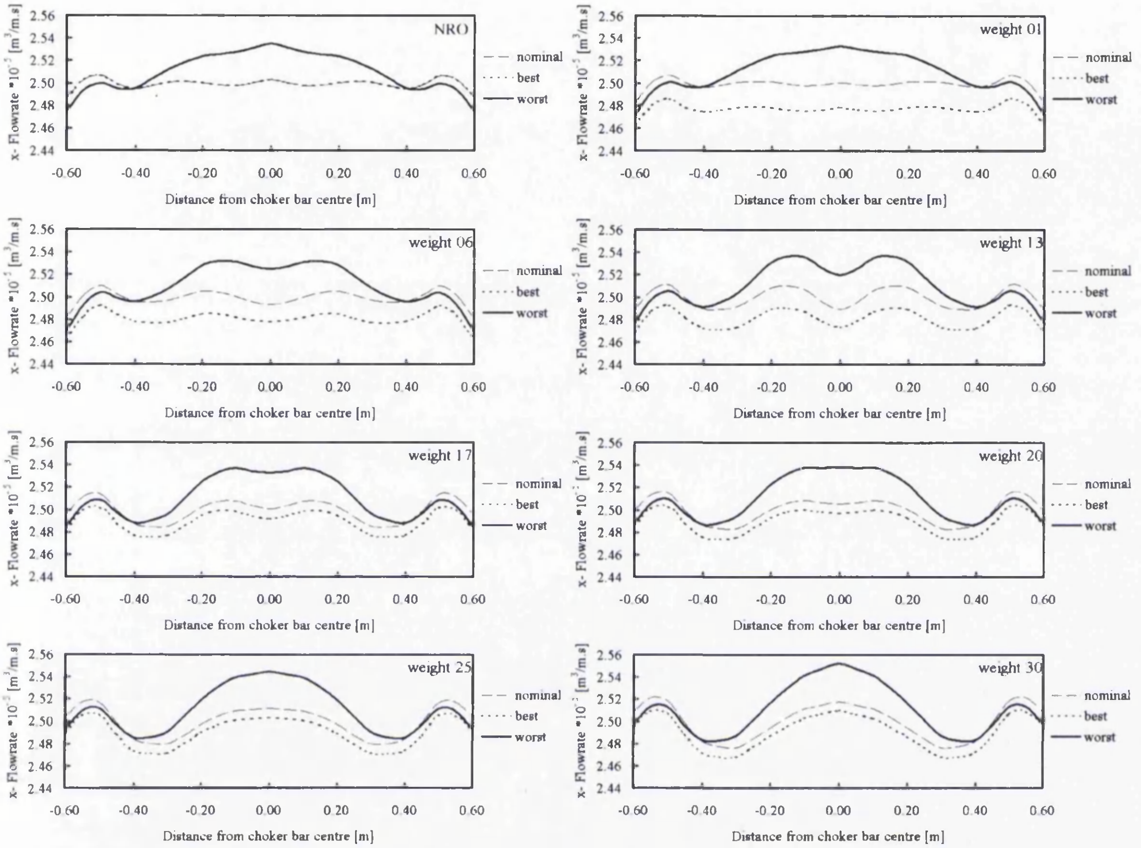


Figure 5.27: Worst, best and nominal exit flow-rates for NRO and several solutions on the 100-sample 'Pareto' curve in Figure 5.22

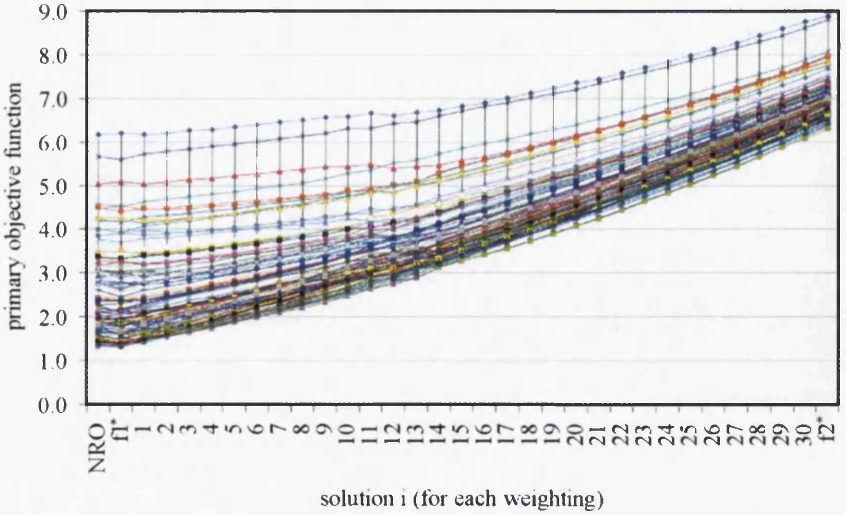


Figure 5.28: Performance curves for each sample for each of the 30 Pareto solutions, the utopia points f_1^* and f_2^* for the 100-sample problem and the deterministic solution, NRO

The central-composite DoE, discussed in Chapter 3, was used to study the variation of the flow uniformity (Equation 5.8) caused by varying the noise factors between the tolerances given in Table 5.3. The study was repeated for each of the 30 Pareto solutions, the utopia points f_1^* and f_2^* for the 100-sample problem and the deterministic solution, NRO. Figures 5.29 and 5.30 show the results of the DoE studies, with Figure 5.29 showing the results in separate plots and Figure 5.30 combining all of the results into a single plot.

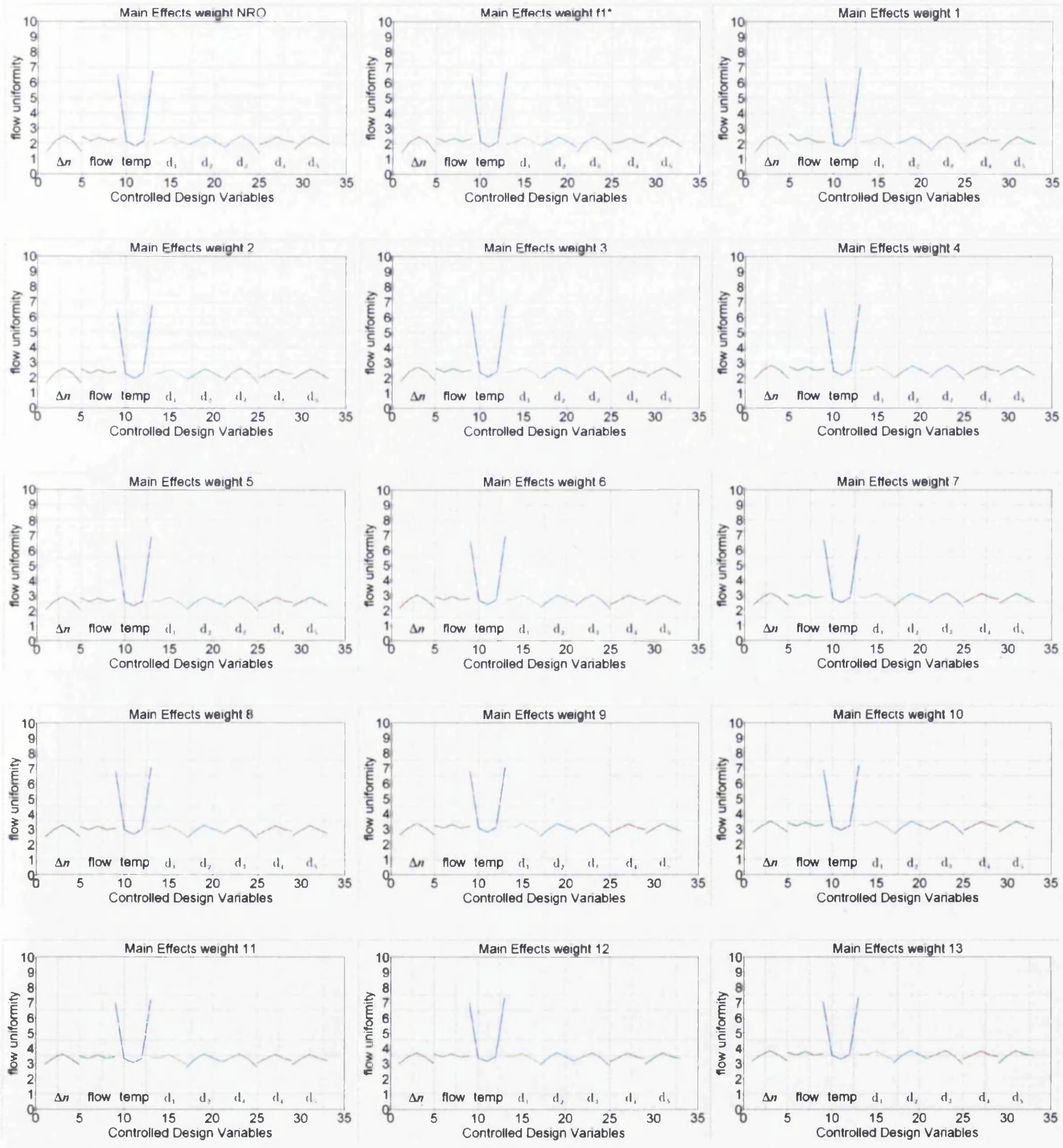


Figure 5.29: Plots showing the effect on the flow uniformity (Equation 5.8) caused by varying the noise factors using the Central Composite DoE at each of the 30 Pareto solutions, the utopia points f_1^* and f_2^* for the 100-sample problem and the deterministic solution, NRO

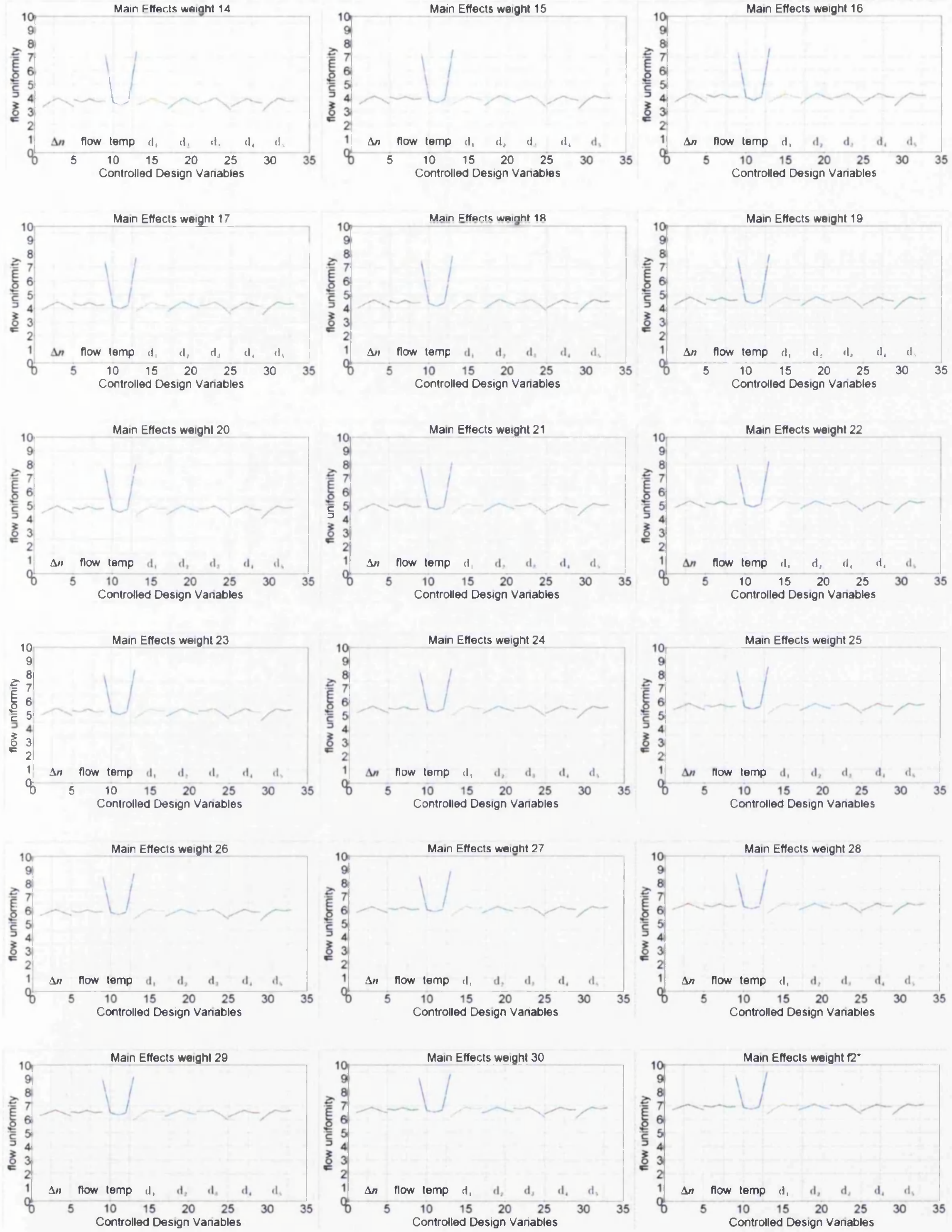


Figure 5.29: ***(continued)***

The overall trend in Figure 5.29 is, that of all the noise parameters, the greatest effect on the performance or flow uniformity (Equation 5.8) is caused by the variation of the operating temperature. It can be seen, that moving from f_1^* to f_2^* the variation caused by the temperature decreases and as would be expected from the problem formulation, the variation in performance decreases and the mean performance increases. The effect of the variation of the other noise parameters can be more easily seen in Figure 5.30. It can be seen that as one moves from f_1^* to f_2^* the mean performance increases and the effects of the noise parameters on the variation of performance are as follows:

- decreases for Δn.
- Negligible change for the input flow-rate.
- For d_1 it is roughly the same up to weight 13 and thereafter the same as weight 14.
- Negligible change for d_2.
- Slight decrease in variation for d_3.
- Negligible change for d_4.
- Slight decrease in variation for d_5.

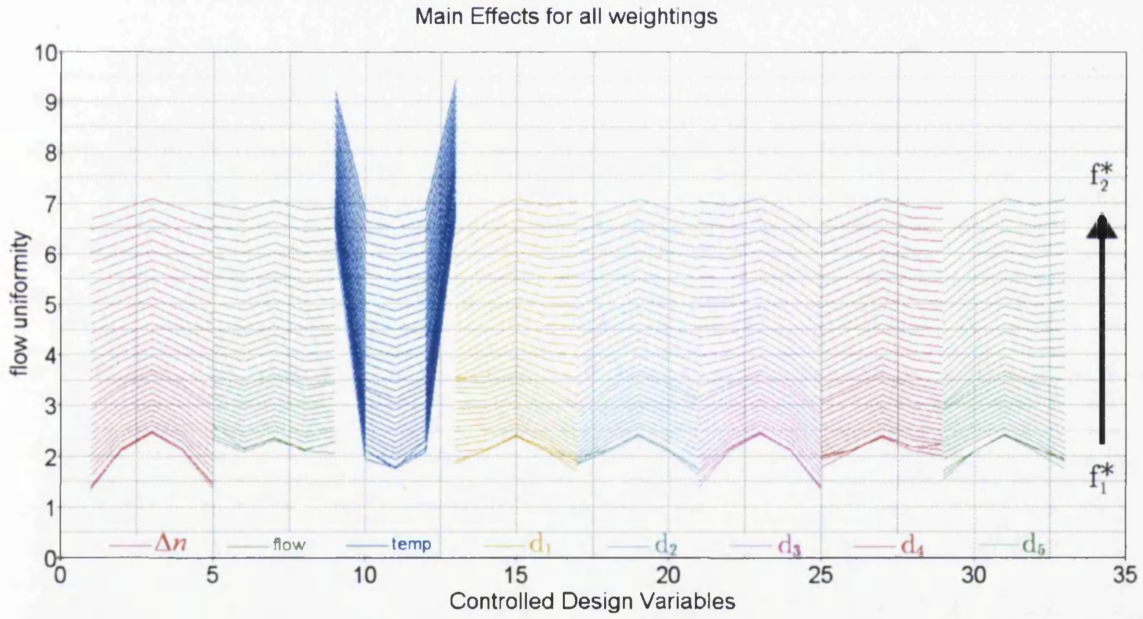


Figure 5.30: Figure 5.29 combined into a single plot

At this stage of the work, the fact that the temperature has more influence on the mean and variation than the other noise variables is not so important, since the aim here is to develop the methodology through the use of the slit die application. The variation of temperature was assumed at the beginning of the study and in further implementations of the methodology the assumptions on the variations of the parameters needs to be carefully considered using a DoE study in the manner used above.

Choosing which solution is the robust design is not as straightforward as the 10-sample problem, where the selection was obvious. The selection has to be decided by the manufacturer, based upon a compromise between what variation in the thickness of the plastic sheets is tolerable and what predictability in the variation is required.

If the manufacturer wants high predictability then, by the trend in the shape of the choker bar in Figure 5.25 and the variables in Figure 5.26, the use of a choker bar becomes more and more unnecessary as the amount of predictability required increases but the performance decreases. Figure 5.31(a) shows the expected performance scatter without the choker bar and Figure 5.31(b) shows the corresponding ‘best’, ‘worst’ and ‘nominal’ flow-rate distributions.

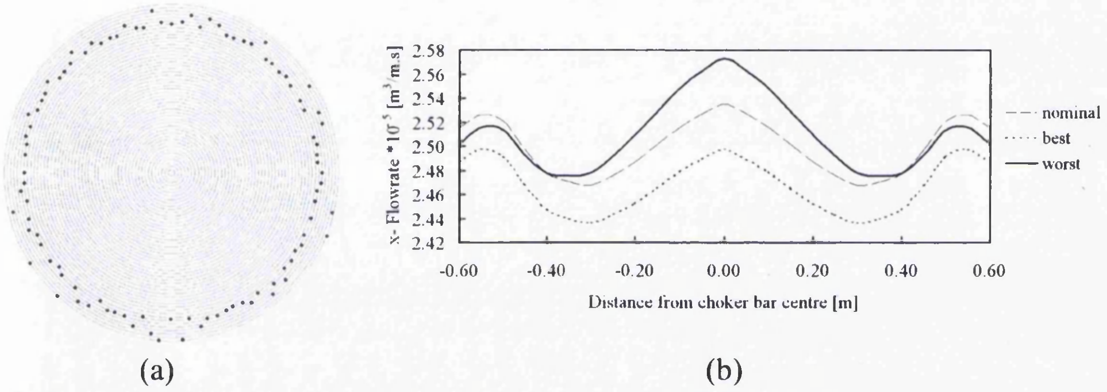


Figure 5.31: Performance for a fully open flow channel with variations of temperature, input flow rate and Δ*n*; (a) performance scatter (with the same scale as the plots in Figure 5.23); (b) worst, best and nominal exit flow-rates

If the manufacturer wants optimal mean performance, whilst accepting the inherent high unpredictability in the thickness distribution, then the NRO or f_1^* solutions are the best solutions to choose. So overall, if the designer wants high predictability and uniform sheets then something else other than using a choker bar has be done in the changing the process. This will require tightening the tolerances on the noise parameters (Table 5.3) since the variation in the temperature, input flow rate and Δ*n* will still have a significant effect on the flow uniformity. This can be seen in Figure 5.31 showing, for a fully open flow channel, the variation in the flow uniformity caused by variations of temperature, input flow rate and Δ*n*.

Assuming that the results remain a true representation of the system then from the DoE study above it can be seen that, of all the noise factors, reducing the operating temperature variation will have the greatest effect on changing the flow uniformity. So, by reducing the variation of operating temperature, the variation of performance of the NRO or f_1^* solutions will be more predictable and hence more feasible. Further investigation is required to find the acceptable variations of the system parameters. This could be done by using the deterministic optimum as the choker bar design and then determine the optimum values for the tolerances on the noise factors to bring about the minimum mean and variation of performance. This would also require factoring in the cost of meeting the tolerance specifications into the objective functions.

5.7 CONCLUDING REMARKS

- In this chapter both deterministic optimization and robust design optimization have been applied to the design of choker bars for use as a flow restricting mechanisms in a 1.2m wide, high-pressure slit die. The optimization has been used to determine the choker bar profiles required to give the best possible exit flow uniformity obtainable.

- For the particular die analyzed and from the deterministic optimization, increasingly sharp variations in the imposed restriction are required across the width of the die as the melt becomes more shear thinning. For the most extreme case, the choker bar profile fully spans its move limits, indicating that it is close to the limits of its effectiveness in bringing about an acceptably uniform flow distribution.

- The optimization capability allows correct initial set up of the flow restrictor, reducing or eliminating the run-in time required for adjustment. Importantly, it also shows how effective the choker bar adjustment can be in a given application, and whether the best flow uniformity achievable is acceptable. Thus the practical operating range of a proposed die design, in terms of material properties and flow rates, can be determined in advance of practical trials.

- It will be necessary, in some cases at least, to machine a choker bar to the computed optimum profile, with the possibility of minor adjustments on-line by bending. This, in turn, suggests the use of interchangeable choker bars, optimized and machined for specific applications, thus providing the means to extend the operating range within which a die can produce uniform flow distribution.

- There is a compromise between the quality of the results and the time taken to reach the result when choosing the appropriate optimizer. The GA finds better solutions than both DOT and MARS, but requires 10-times more function evaluations to reach the solution. The compromise is not necessarily based on which optimizer to use according to how expensive the simulation is, but could also involve using a hybrid of the methods, for example using the GA to locate the region of the optimum, and then use a gradient-based method to quickly 'zoom-in' on the optimum solution. Therein lies the problem of when to switch from the GA to the gradient-based optimizer. Some work has been carried out in this direction in [5.22].

- The robust design formulation has been developed through the use of a sample size of 10 with a uniform distribution of the noise parameters. The use of this small sample size enabled rapid assessment of several multi-objective methods and a new method based called PFM-PP was developed. For the set-up of the methods in this work and for the current problem, PFM-PP generates a uniform distribution of points resulting in a smooth Pareto front. It also requires five-times fewer function evaluations to generate the solutions than the current NBI and PP implementations.

- Using the procedure developed in the 10-sample problem together with PFM-PP a more realistic and computationally more expensive problem was studied using a normally distributed sample of size 100.

- A question arising from the comparison of the 10-sample results to the 100-sample results is: What sample size will give the best approximation of the mean and variance of the population? Clearly the sample of size is 10 is an inadequate representation of the population and the sample size of 100 may also be inadequate. The appropriate size needs to be determined by further research investigating the results for other sample sizes. For this work the sample size is assumed to be appropriate.

- If only one CPU is available then the computational cost in terms of CPU time of conventional deterministic optimization as opposed to stochastic robust design optimization is proportional to the size of the sample used. If the sample size is 1000 then 1000 simulations are required for the stochastic optimization compared to one simulation for the deterministic optimization. Thus the stochastic optimization would take a prohibitive amount of time. To address the problem, there are two possible solutions:
  (i) If the number of CPUs available equals the sample size then the samples can be all run in parallel meaning that the effective wall-clock time evaluation of performance is the same for both the deterministic optimization and the stochastic optimization. This situation is becoming increasingly feasible through the availability of cheaper and faster multi-processor machines.
  (ii) If an accurate approximation of the design space incorporating the variation caused by the noise factors is available, then CPU time is no longer an issue, but generating an accurate approximation is non-trivial (see Chapter 9 for further remarks on approximations and CPU issues).

- The aim here was to develop a RDO formulation through the use of the slit die application. The significance of the tolerances on the noise parameters was not addressed until the end of the study and instead experience-based assumptions on the tolerances were made. Assuming that the tolerances used are 'optimal' the results of the RDO study provide the designer with a set of possible solutions reflecting the compromise between predictability and performance.

- The 100-sample study highlighted the importance of looking at factor significance at the beginning. It became apparent that one factor, temperature, dominated the response variations of the parameters meaning that the influence of the other noise parameters were insignificant in comparison. This brings about some possibilities for modifications to the current RDO formulation:
  (i) Determine the optimum values for the tolerances on the noise factors to bring about the minimum mean and variation of performance. This would also require factoring in the cost of meeting the tolerance specifications into the objective functions.

(ii) Assuming that the tolerances are optimal then some of the noise factors can be eliminated from the study. In other words a DoE study is required initially to eliminate the insignificant noise factors. This is the approach used by Chen in [4.89]. Then optimization is required to determine the tolerances on the variation of the noise parameters with the constraints being the cost of tightening or relaxing them.

5.8 REFERENCES

[5.1] Bates S.J., Sienz J. Pittman J.F.T., Langley D.S., "Enhancing Slit Die Performance by Optimization of Restrictor Profiles", J. Polymer Engineering and Science, Aug. 2003.

[5.2] Winter H.H., Fritz H.G., "Design of dies for the extrusion of sheets and annular comparisons: the distribution problem", Poly Eng Sci, 26, pp.543-553, 1986.

[5.3] Toombs R.S., 'Computer Aided Design of Extrusion Dies', M. Eng Dissertation, Dept. Chemical Engineering, University of Wales Swansea, UK, 1994.

[5.4] Wortberg J., Tempeler K., Kunststoffe, 73, 404 (1983).

[5.5] Gross H.G., Michaeli W., Pohler F., Ulrich J., Zeitschrift Kunststoffe, plast europ, 84, 20 (1994).

[5.6] Sander R., Pittman J.F.T., Poly Eng Sci, 36, 1972 (1996)

[5.7] Gifford W.A., ANTEC 1999 - Annual Technical Conf, Soc. Plastics Engineers USA, 262 (1999).

[5.8] Smith D.E., Tucker III C.L., Tortorelli D.A., "Simulation of Materials Processing: Theory, Methods and Applications", (edited Shen and Dawson), Balkema, Rotterdam, 619, 1995

[5.9] Smith D.E.,. Tortorelli D.A., Tucker III C.L, ANTEC 95 - Annual Technical Conf, Soc. Plastics Engineers USA, 42 (1995).

[5.10] Idem, Computer Methods Appl Mech. and Engrg, 167, 283 (1998).

[5.11] Idem, Ibid., 167, 303 (1998).

[5.12] Na S.Y., Lee T-.Y., HPC Asia 97 - High Performance Computing Conf, Seoul, S. Korea, IEEE, Ch 137, 601 (1997).

[5.13] Wang Y., Simulation of Materials Processing: Theory, Methods and Applications, (edited Huetink and Baaijens), Balkema, Rotterdam (1998), 423.

[5.14] Pittman J.F.T., Sander R., Int Poly Proc, 9, 326 (1994).

[5.15] Sander R., "Modelling of Slit Die Extrusion", PhD Thesis, University of Wales Swansea, 1994.

[5.16] Miller C., Ind Eng Chem. Fundam, 11, 524 (1972).

[5.17] Hinton E., Owen D.R.J., Finite Element Software for Plates and Shells, Pineridge Press Ltd, Swansea, UK (1983).

[5.18] Pittman J.F.T., Sander R., Schuler W., Pick H., Martin G. Stannek W., Int Poly Proc, 10, 137 (1995).

[5.19] Vanderplaats Research & Development Inc, http://www.vrand.com, 2003

[5.20] 'Altair HyperWorks Version 5.1', Altair Engineering Limited, http://www.uk.altair.com, 2002.

[5.21] Personal communication with J.F.T Pittman, Uni. of Wales Swansea, 2001.

[5.22] D.S. Langley, "Genetic algorithm focussed comparative optimization studies for a broad scope of engineering applications", PhD Thesis, University of Wales Swansea, 2003.

CHAPTER 6

A-PILLAR DESIGN USING COMPUTER-BASED OPTIMIZATION, MANUFACTURE CONSTRAINTS AND CRASH ANALYSIS

Summary: A series of computational tools have been used to fully design a virtual component. The process uses topology optimization, manufacturing simulation, shape optimization and crash simulation. The component designed is a cast aluminium A-pillar for the Lea-Francis sports car that meets Federal Motor Vehicle Safety Standard (FMVSS) 216, Roof Crush. The design was achieved with the use of a low-cost holistic virtual design process. This design process used the topology optimization tool in Altair OptiStruct to generate an initial optimum topology for the A-pillar. Based on computer manufacturing simulations, experience, and the topology optimization results, an initial concept design was produced. The performance of this initial design was assessed using implicit LS-DYNA. The model was then parameterized to enable final shape optimization. Finally, a study into the effect of using a finer mesh density and using different minimum member sizes for the topology optimization is carried out. The results show that the material distribution is dominated by the minimum member size constraint. The effect of using a finer mesh, and a smaller minimum member size is to generate a more complex and different material distribution.

6.1 INTRODUCTION

This chapter is set out as follows:

Section 6.1: Gives the background to the problem.

Section 6.2: Describes the loading conditions for the roof crush test and defines the available package space.

Section 6.3: Topology optimization is used to generate concept designs.

Section 6.4: The concept design is assessed for manufacturing feasibility and a baseline design is generated.

Section 6.5: Non-linear analysis of the baseline design is carried out to verify that the design meets the roof crush requirements. Then, the design is parameterized and shape optimization is carried out.

Section 6.6: Considers the questions: What is the effect of using a finer mesh, and hence a smaller minimum member size, on the final material distribution? How does the unaltered topology perform in the roof crush test compared to the baseline solution developed in the preceding sections?

Section 6.7: Gives some brief concluding remarks.

Section 6.8: Lists the references used in the chapter.

Sienz [6.1] identifies that the 'main deficiency has been and to some extent still is that the (optimization) methods have been poorly integrated into the overall design process.' This chapter addresses this issue and is an industrial example where optimization is the driving force behind the holistic virtual design of an automotive component. The aim of this chapter is to use a series of computational tools and explore the feasibility to fully design a virtual component. The process uses topology optimization, manufacturing simulation, shape optimization and crash simulation. The novelty of this approach is that it brings together a variety of computational design tools into a single design process, thus providing an initial feasible design without the need for prior physical prototyping and testing, and thereby reducing time and costs. Typical costs savings are a 20 to 30% decrease in the time to reach a manufacturable solution. The greatest savings, however, come from a reduction in the number of iterations to reach the solutions. Each iteration or refinement requires the manufacture and testing of the component. Using the approach implemented here, a reduction from up to 100 refinements to just one or two refinements can be achieved. This is important for small companies, especially in a highly competitive industry such as the automotive industry. In 1895 the Lea-Francis motor company was established, and launched its first car in 1903 (Figure 6.1(a)). Since the 1950's Lea-Francis has been relatively inactive, lacking the investment required to compete with the larger manufacturers that were investing in cost-effective mass production designs. A new Lea-Francis sports car is currently being developed. A marketing prototype of the car has been built (Figure 6.1(b)) and the current styling has been further advanced (Figure 6.1(c)). The company is very small, aiming at the low-volume niche market, and therefore cannot support the design overhead carried by larger car companies whilst achieving the desired return on investment. To survive, the company must avoid the costs associated with mass-market production, and also the inaccuracy and unreliability associated with niche market design. As a necessity therefore, the company decided to be an early adopter of new technology in order to meet these challenging objectives. This chapter shows how a holistic virtual design approach has been applied to the design of the A-pillar of the latest vehicle. Figure 6.1(c), shows the complete car, and (d), the location of the A-pillar to be designed.

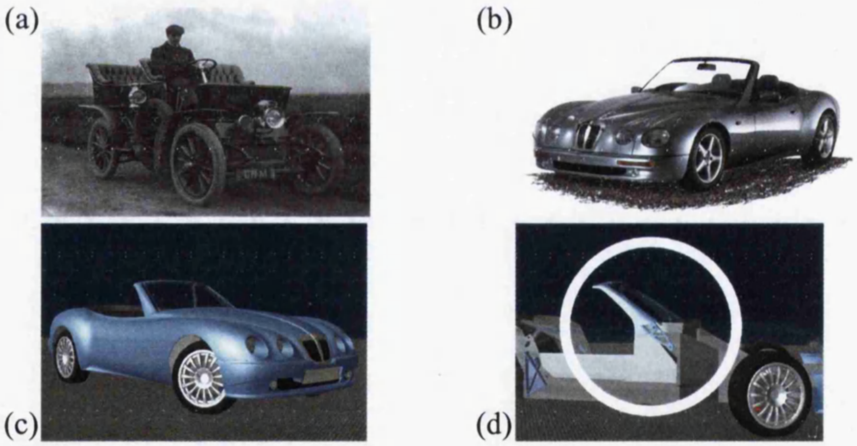


Figure 6.1: (a) First Lea-Francis 1903; (b) Lea Francis prototype model; (c) Virtual image of the latest Lea-Francis prototype; (d) A-pillar location

For convertible cars, the A-pillar design can be one of the main problem areas. The A-pillar is the confluence of many areas of the car, and in durability tests the A-pillar area is often the first to fail. Traditionally, A-pillar design uses steel, which leads to unwieldy designs that are often "patched-up" to meet requirements (Figure 6.2(a)). Also, steel solutions require dedicated tooling, which is undesirable for a small company. Aluminium is a good alternative; a typical aluminium design (Figure 6.2(b)) consists of extruded members welded together. Research has demonstrated the availability of high strength aluminium alloys with yield stresses of around 400 MPa with 15% elongation [6.2]. There is little benefit in using high strength aluminium in a welded structure, because the welds will only be as strong as the basic aluminium used for the welding (unless expensive post-treatment is done). However, a cast aluminium solution can take full advantage of the high strength alloys that are available. Therefore, a casting solution using the high strength aluminium is used for the design.

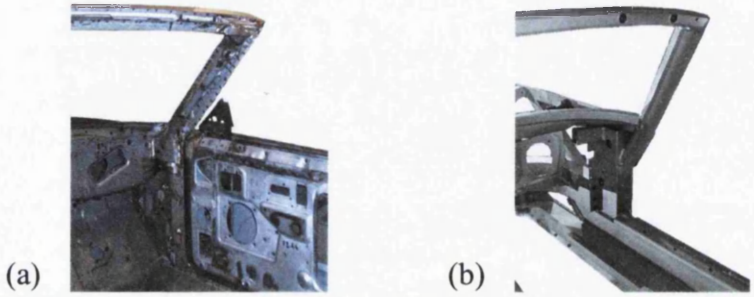


Figure 6.2: Typical A-pillar designs using (a) Steel extrusions; (b) Aluminium extrusions

The objective for this study is to minimize the mass of the A-pillar casting, whilst:

(i) Passing FMVSS 216 roof crush test requirements (see Section 6.2.1).

(ii) Ensuring that the resonant frequency of the windscreen and A-pillar structure is above 20 Hz in order to avoid the impression of "scuttle shake".

As the objectives require high A-pillar stiffness, and the roof crush requirement is the most difficult to meet, optimization is based on the roof crush load case and resonant frequency requirements are checked *a posteriori.*

6.1 PROBLEM DESCRIPTION

6.2.1 ROOF CRUSH TEST

The roof crush test is based on the US Federal Motor Vehicle Safety Standards (FMVSS) number 216 Ref. [6.3]. It defines the loading conditions that apply to A-pillar design. The loading device is an angled rigid plate that is lowered onto the A-pillar. The loading applied to this plate is increased up to a value of 1.5 times the unladen vehicle weight so that

$$P_{r_{req}} = 1.5 \times g \times m, \qquad (6.1)$$

where $P_{r_{req}}$ is the required reaction force, g is the acceleration to gravity and m is the unladen vehicle mass. For the Lea Francis car $P_{r_{req}}$ =16,995N. The load case is achieved by moving the plate at 13mm/s along the prescribed direction until the peak load is achieved. During this time the displacement of the rigid plate along its path must not exceed 127mm. The test device is oriented so that:

(a) "Its longitudinal axis is at a forward angle (in side-view) of five degrees below the horizontal, and is parallel to the vertical plane through the vehicle's longitudinal centreline."

(b) " Its transverse axis is at an outboard angle, in the front view projection, (shown in Figure 6.3(b)), of 25 degrees below the horizontal." [6.3]

Typical results are schematically represented in Figure 6.3(a). The green line shows a successful design where the force requirement is met before the displacement limit is reached, whereas the red line reaches the displacement limit first, and therefore represents a failed design.

In the case of a cast A-pillar, the most likely mode of structural failure would be for the casting to fracture before the FMVSS force limit has been reached. Consequently, a further constraint is put on the design: from Ref. [6.4] the component must have no plastic strain greater than 10% at the FMVSS peak force levels.

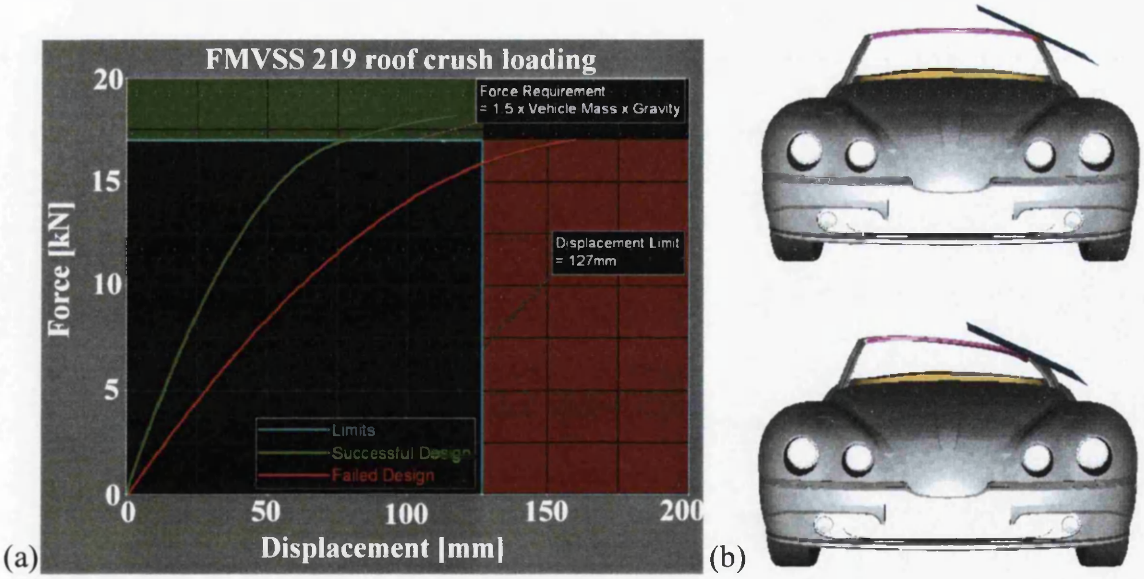


Figure 6.3: Schematic representation of FMVSS 216: (a) results for typical failed and successful designs; (b) rigid loading plate position and orientation

6.2.2 Package space

The package space available is shown in Figure 6.4 as the yellow region. The non-designable space consists of the attachments, the outer A-surface, the unloaded side (all red), and the body structure (grey). The assumption of making the unloaded side as non-designable was deemed satisfactory after performing preliminary topology optimization. The total mass of the package space material is 12.58kg. The roof crush loading is represented as a single static load case, applied as nodal forces in the direction of the movement of the plate.

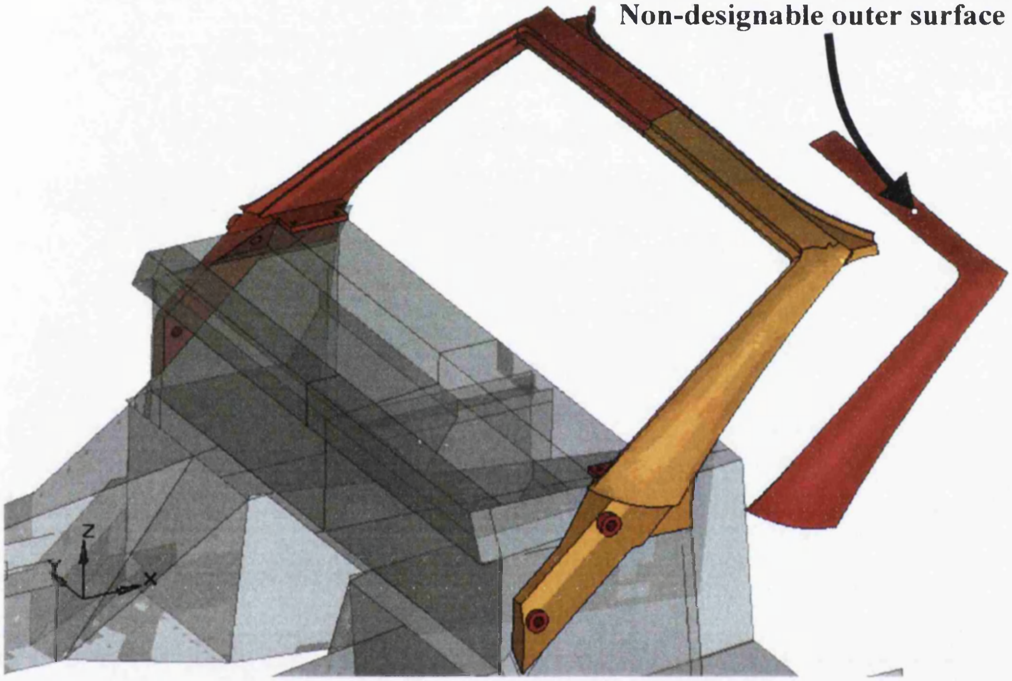


Figure 6.4: Package space identification (designable (yellow), non-designable (red) and the body structure (grey))

6.2.3 Overall design process

The overall design procedure is as follows: Firstly, the problem and package space is identified and this is used to carry out concept topology optimization with manufacturing constraints, to identify the important structural load paths. Next, an initial design is generated based upon the concept designs. This is assessed for manufacturing feasibility using computational simulations of the manufacturing process. Based upon the manufacturing assessment, a 'baseline', manufacturable design is generated. The performance of the baseline design is then assessed using non-linear

FE analysis to simulate the roof crush test and eigen value analysis to identify the natural frequency modes. Finally, the mass of the baseline design is further reduced using shape optimization. A diagram of the overall design process is shown in Figure 6.5.

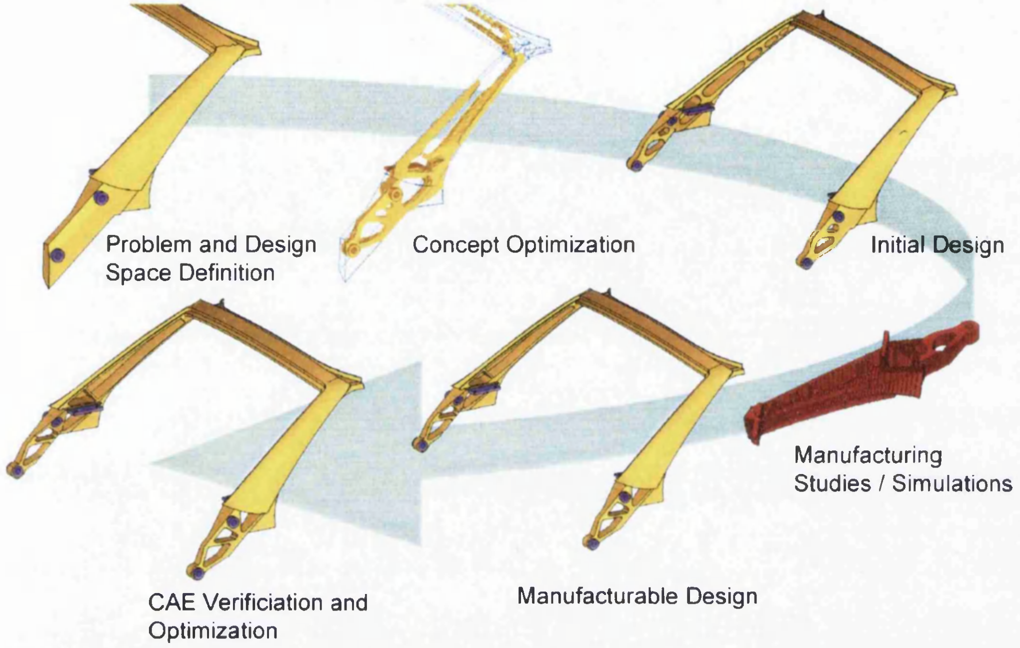

Problem and Design Space Definition
Concept Optimization
Initial Design
Manufacturing Studies / Simulations
Manufacturable Design
CAE Verificiation and Optimization

Figure 6.5. The overall design process

6.3 CONCEPT OPTIMIZATION

In this section a concept design is generated using the topology optimization software OptiStruct [6.5]. The package space is shown in Figure 6.4, and it is the starting point for the optimization. The use of the manufacturing constraints, discussed in Section 6.3.1, is the driving criteria for the topology optimization. So, the density method is used for the topology optimization (Chapter 2) because it is the only method currently implemented in OptiStruct for use with the manufacturing constraints option.

6.3.1 CONCEPT STRUCTURAL ARCHITECTURES

The objective of the optimization is to maximize the stiffness of the A-pillar, subject to a design constraint, which is specified as a target mass. Since the applied loading is fixed, to maximize the stiffness requires minimizing the compliance of the structure. The target mass is defined as the fraction of solid material to remain, e.g. a target mass of 0.3 means that 30% of the solid material is to remain.

The A-pillar is to be designed as a casting. Manufacturing constraints can be applied in OptiStruct by using draw direction constraints, where the material layout produced can be orientated to a particular direction [6.6]. Two draw directions are available; one accounting for castings where the mould is extracted in one direction, and one for a split casting where the moulds are removed from either side of the casting. The effect of these constraints is that only cavities that are open and aligned with the sliding direction of the die are permitted in the optimization. An indication of the FE model can be seen below in Figure 6.7 and the topology optimization problem can be stated as:

$$\begin{aligned}\text{minimize:}\ & \text{Compliance}\\ \text{subject to:}\ & \text{Draw direction constraints}\\ & \text{Various target masses}\end{aligned}\tag{6.2}$$

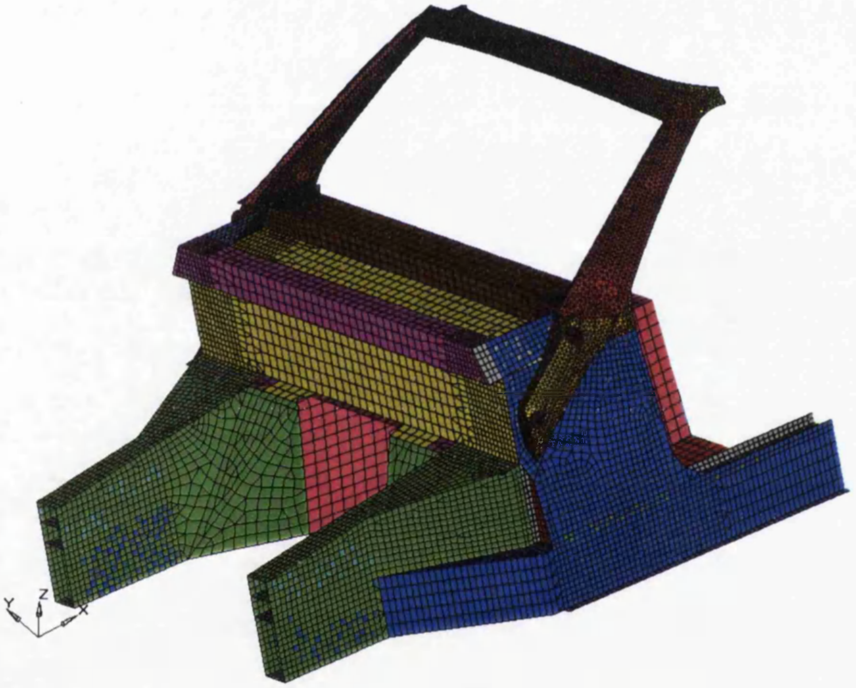
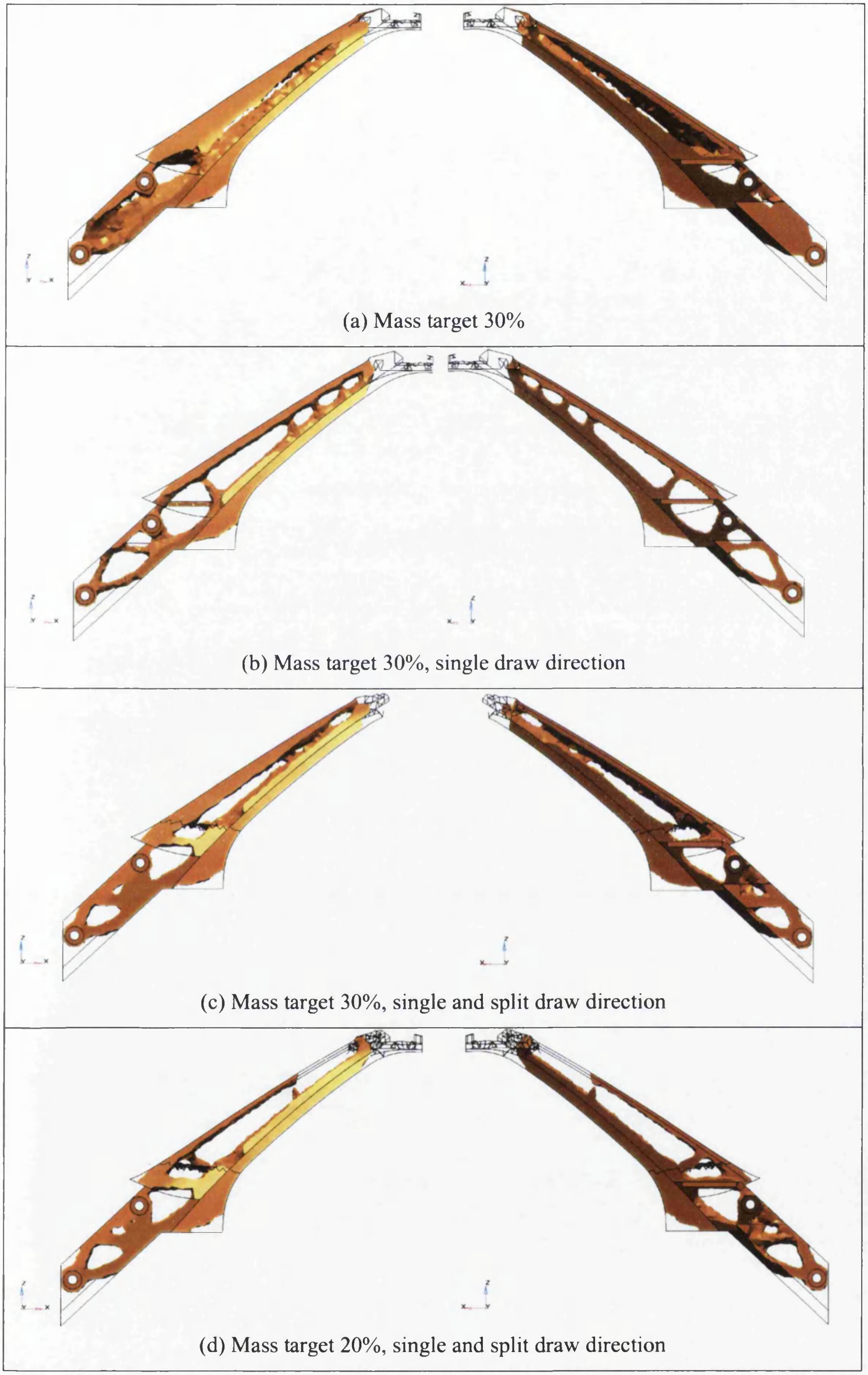


Figure 6.6: FE mesh of the model used for the topology optimization

The full model consists of 74,510 elements and 230,838 degrees of freedom. The design domain comprises 21,221 elements. It was analyzed to give solutions for various combinations of target mass fractions and manufacturing casting constraints. The results are shown in Figure 6.7. The typical solutions show a definite load path along the bottom edge, from a loading point to the fixings. It should be noted that these results do not show the non-designable A-surface, which also carries load. The typical CPU time to reach convergence is 20 minutes on a 2.6GHz Pentium 4 PC.



(a) Mass target 30%

(b) Mass target 30%, single draw direction

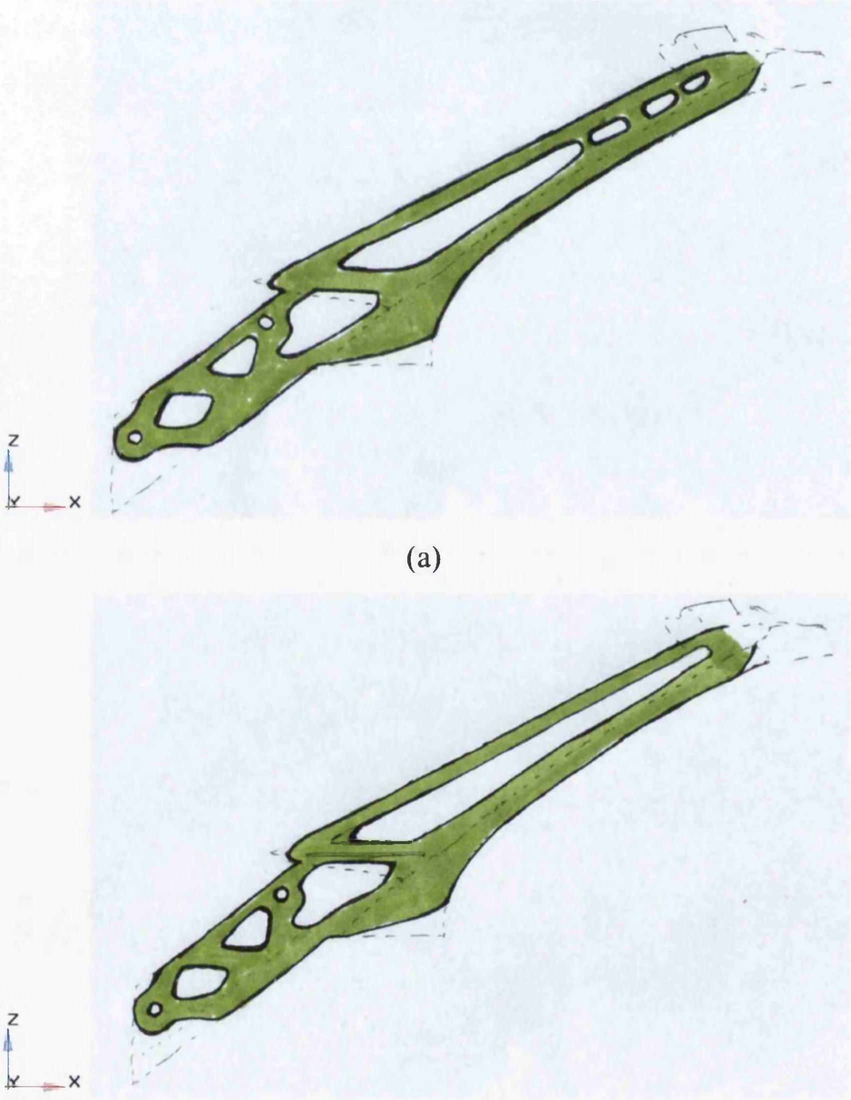
(c) Mass target 30%, single and split draw direction

(d) Mass target 20%, single and split draw direction

Figure 6.7: Topology optimization results for various combinations of target mass and draw direction constraints

6.3.2 Concept Sorting

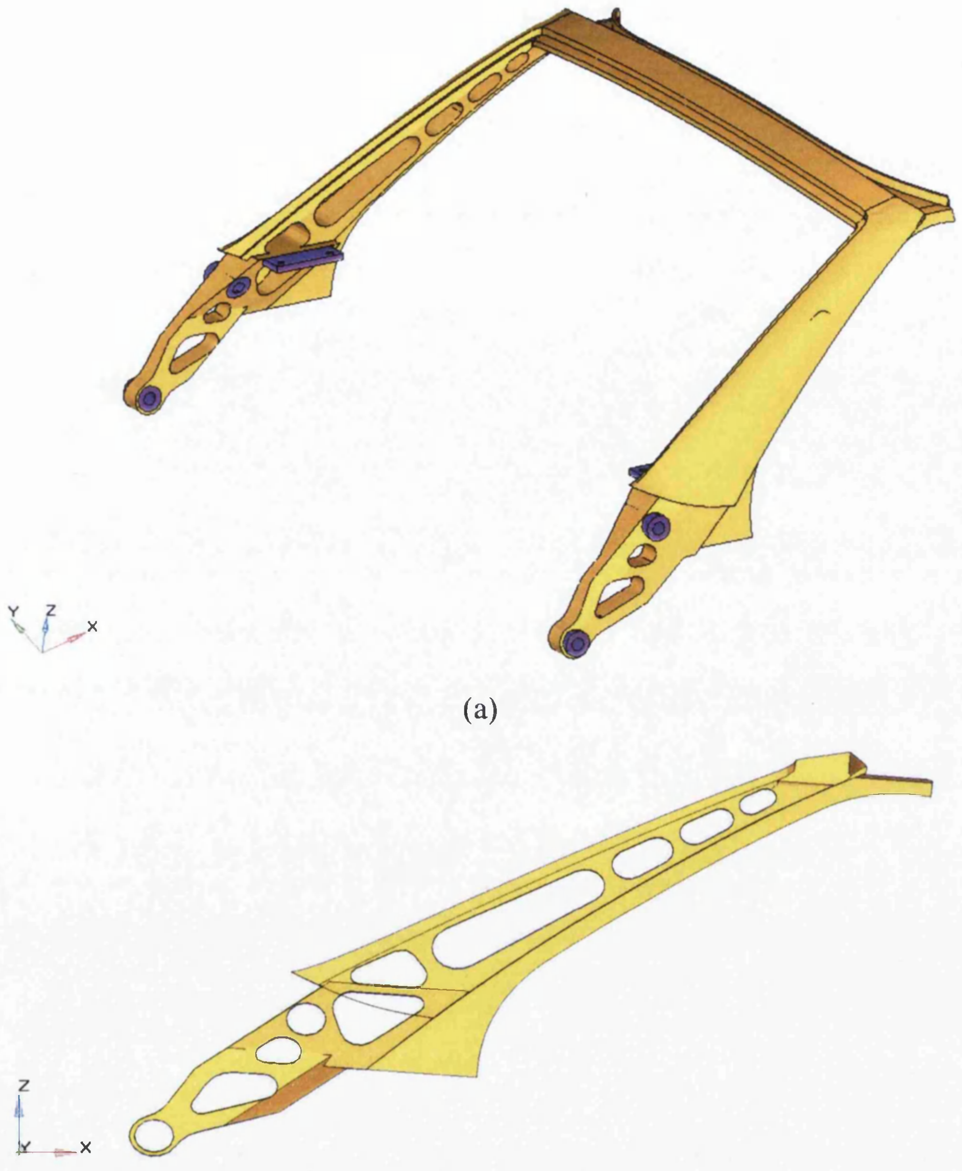
From the topology results in Figure 6.7, the next stage is to sketch interpretations of the design concepts suggested by the structure's natural performance. Two sketches of possible designs can be seen in Figure 6.8; they are generally the same except, with the only difference being, that the design shown in Figure 6.8(a) has three ribs at the top of the section. Since this is a virtual design process, there is only a small overhead in running both these design concepts through the rest of the process. Using Altair OSSmooth, the geometry recovery phase of the interpreted design was initially performed on the concept presented in Figure 6.8(a), resulting in the initial concept design shown in Figure 6.9.



(a)

(b)

Figure 6.8: Sketches of the underlying features of the OptiStruct results



(a)

(b)

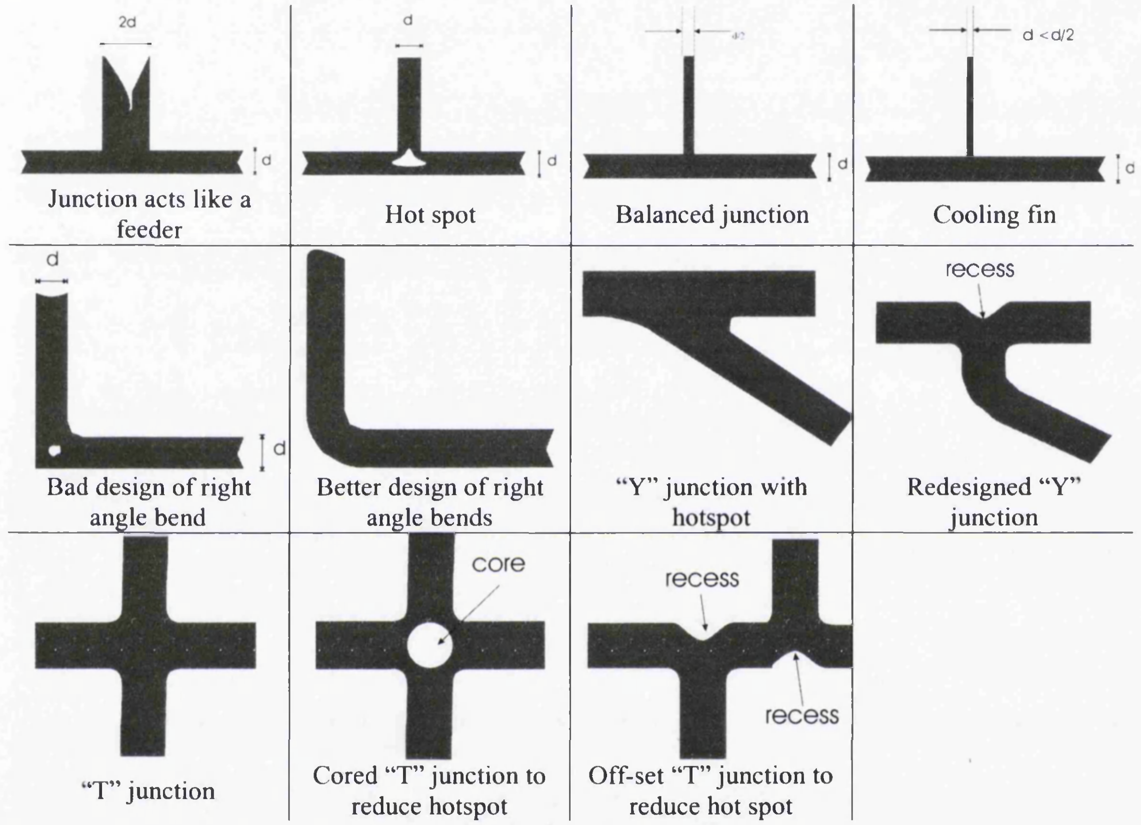
Figure 6.9: Initial concept design: (a) angled view; (b) cross-sectional view

6.4 MANUFACTURE

A fundamental issue to be addressed by any design is manufacturing feasibility, i.e. can the design be made? For the vast majority of manufacturing processes numerical techniques are available to perform a virtual assessment on whether a design is feasible. Such assessments are an absolute necessity for any holistic design process. For the Lea-Francis A-pillar casting design, an initial review was undertaken using established casting design guidelines, followed by a virtual assessment of the casting process using the analysis code MagmaSoft [6.7].

6.4.1 REVIEW AGAINST DESIGN GUIDELINES

The initial OptiStruct concept design was reviewed against experience-based casting design guidelines. Typically, manufacturing guidelines relate to features within a design (e.g. junctions) that are required to be casting-friendly. Casting integrity is mainly affected by shrinkage, so it is generally easier to cast sections that are uniform than those with many changes of geometry. Figure 6.10, adapted from Ref. [6.2], illustrates some 'rules of thumb' for good and bad junction designs.



Junction acts like a feeder

Hot spot

Balanced junction

Cooling fin

Bad design of right angle bend

Better design of right angle bends

"Y" junction with hotspot

Redesigned "Y" junction

"T" junction

Cored "T" junction to reduce hotspot

Off-set "T" junction to reduce hot spot

Figure 6.10: Schematics of good and bad junction design

Consequently, a revised design was produced which involved two major alterations to the initial design:

(i) The uniformity of the cross section was improved. Section A-A in Figure 6.11, shows how the cross-section of the initial design is not well balanced, which could lead to shrinkage problems. A better solution is to improve the uniformity of the cross-section thickness as shown in the revised section A-A.

(ii) The ribs have been removed from the top of the A-pillar, and the other remaining ribs have been made thinner to reduce porosity at the junctions.

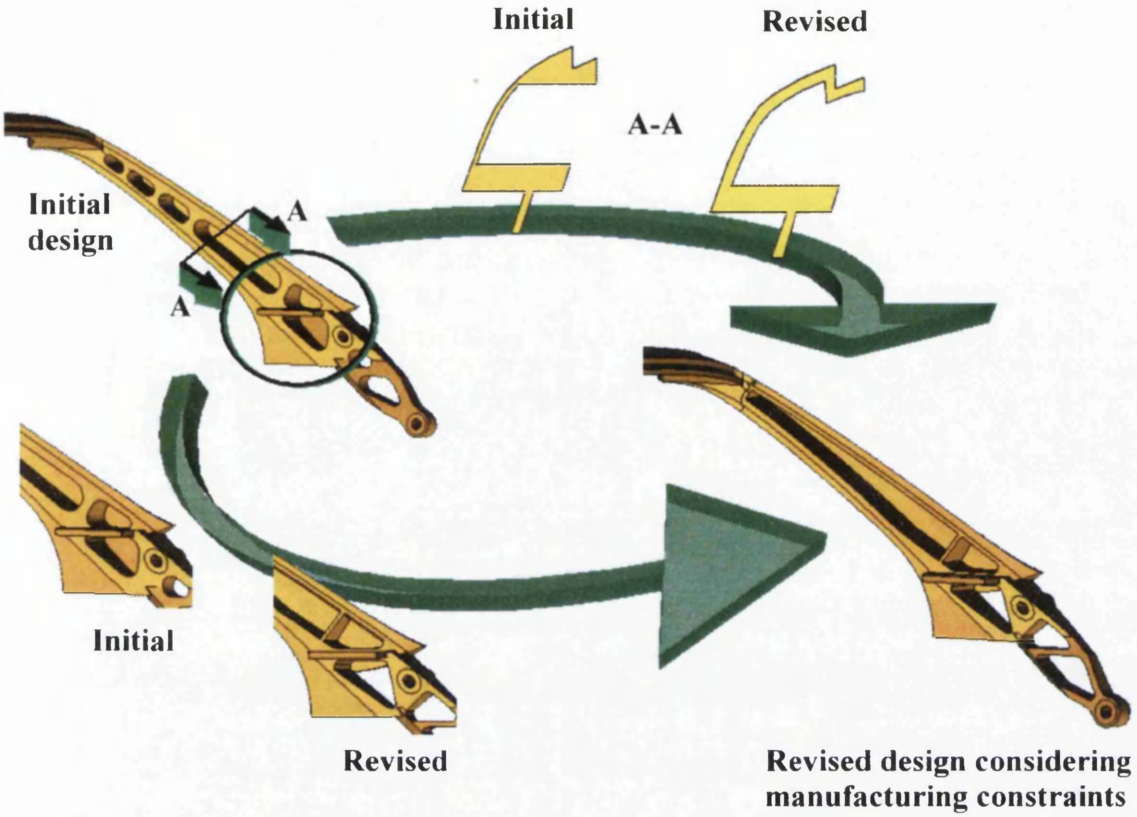


Figure 6.11: Initial manufacturing design input

6.4.2 SOLIDIFICATION AND MOULD-FILLING SIMULATIONS

A solidification simulation using MagmaSoft [6.7] was performed by Turan [6.8], for the initial design in Figure 6.12(a), and the revised design in Figure 6.12(b). This simulation assumes that the mould for the casting is instantly filled with material; this enables rapid assessment of problem areas exhibiting high porosity.

The simulation results for the initial design, Figure 6.12(a), show regions of high porosity, especially around the ribs at the top of the A-pillar and in the walls. The revised design exhibits significantly lower levels of porosity. For example, the porosity levels in the upper wall of the A-pillar shown Figure 6.12(b) are the size of 'pin-pricks' and have negligible impact on the structural performance, and porosity has been reduced at the junctions because of the thinner ribs. The remaining regions of porosity could easily be overcome with the addition of heat sink mechanisms or feed metal reservoirs in the mould.

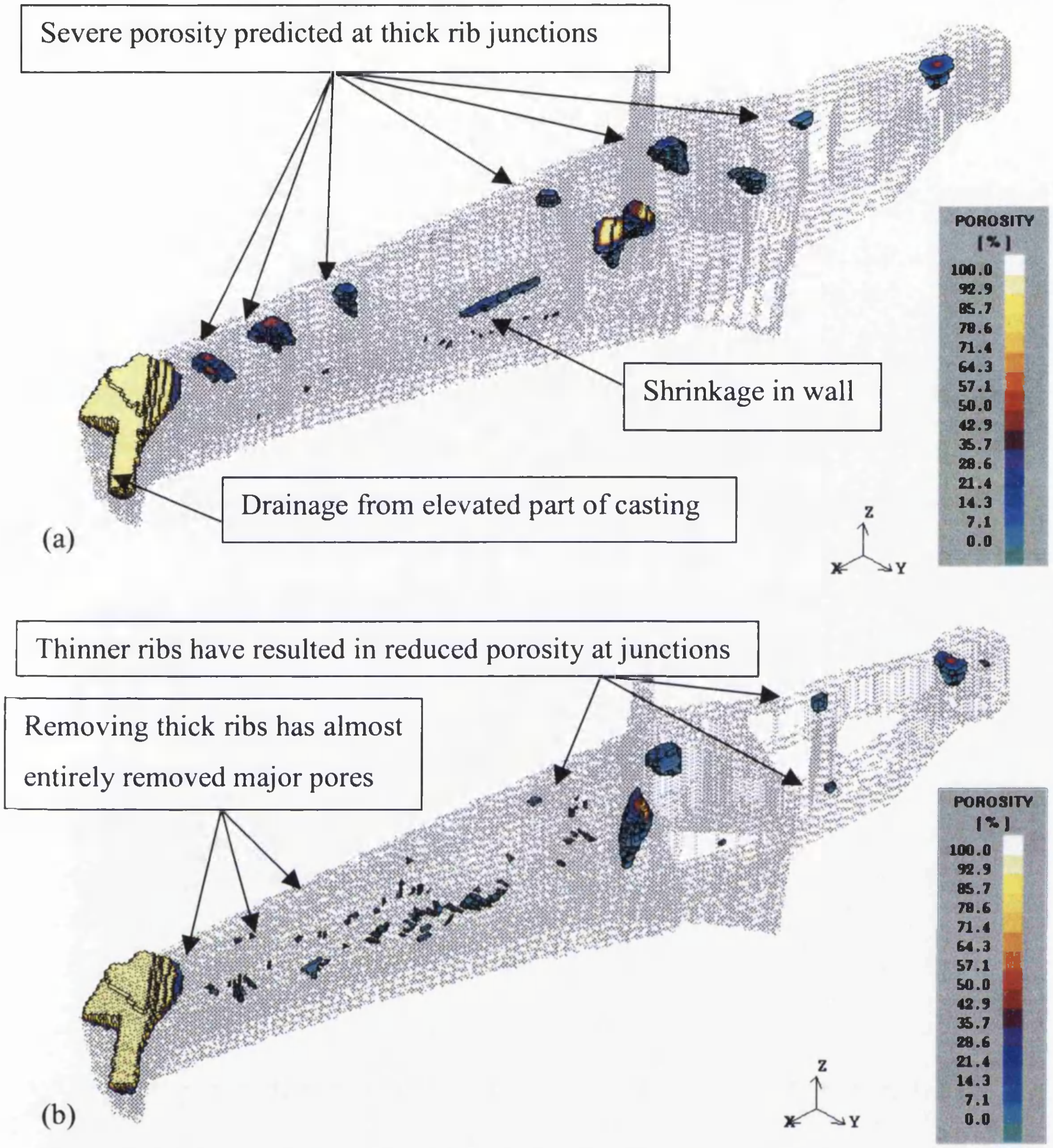

Severe porosity predicted at thick rib junctions
Shrinkage in wall
Drainage from elevated part of casting
(a)
POROSITY
[%]
100.0
92.9
85.7
78.6
71.4
64.3
57.1
50.0
42.9
35.7
28.6
21.4
14.3
7.1
0.0
Z
X
Y
Thinner ribs have resulted in reduced porosity at junctions
Removing thick ribs has almost entirely removed major pores
(b)
POROSITY
[%]
100.0
92.9
85.7
78.6
71.4
64.3
57.1
50.0
42.9
35.7
28.6
21.4
14.3
7.1
0.0
Z
X
Y

Figure 6.12: Porosity results from instant fill solidification simulation for (a) initial design, and (b) revised design

Figure 6.13 shows a simulation of how the mould fills for one in-gate position. The number of in-gates and positions can be optimized at a later stage, but at this stage this simulation gives a quick assessment of the design and may have indicated manufacturing complications with the design. The simulation shows that no 'splashing' of the molten metal has occurred and that the mould fills satisfactorily. A cold front has formed (blue area); this can be overcome by introducing more in-gates. For the A206 aluminium alloy used here the casting, liquidus and solidus temperatures are 720ºC, 650ºC and 570ºC respectively, and the temperature-viscosity curve is shown in Figure 6.14.

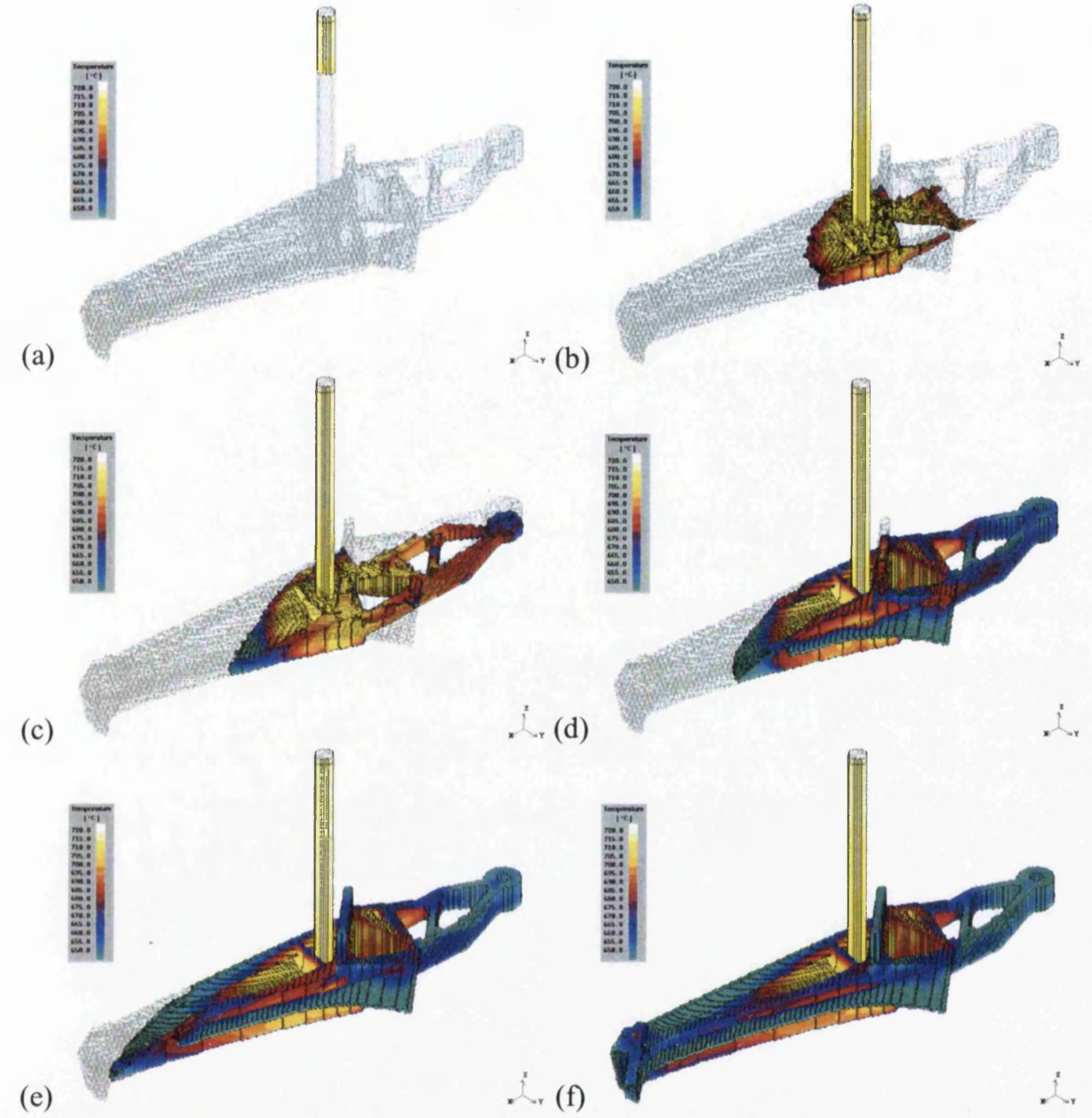


Figure 6.13: Filling simulation of the revised design

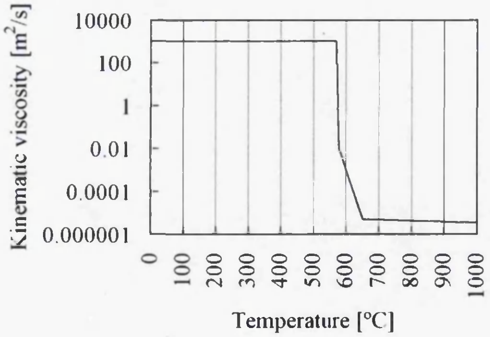


Figure 6.14: Viscosity-temperature curve for the Aluminium alloy A206

The conclusion from this stage of manufacturing feasibility assessment is that there is a high confidence that the revised design will be suitable for manufacture by casting.

6.5 NON-LINEAR ANALYSIS VERIFICATION AND OPTIMIZATION

6.5.1 ASSESSMENT OF THE BASELINE DESIGN

For the final phase of the design process, a non-linear analysis of the baseline design was done in order to accurately predict its performance. This required the explicit modelling of a rigid plate, that deforms the A-pillar, and the resulting non-linear behaviour of the geometry and the material. At this stage the revised design, shown in Figure 6.15, was assessed using the implicit solver in LS-DYNA [6.9], which is well suited to the quasi-static loading regime encountered in the roof crush test. The solution time for this particular problem was on average 15 minutes using a HP J-class machine.

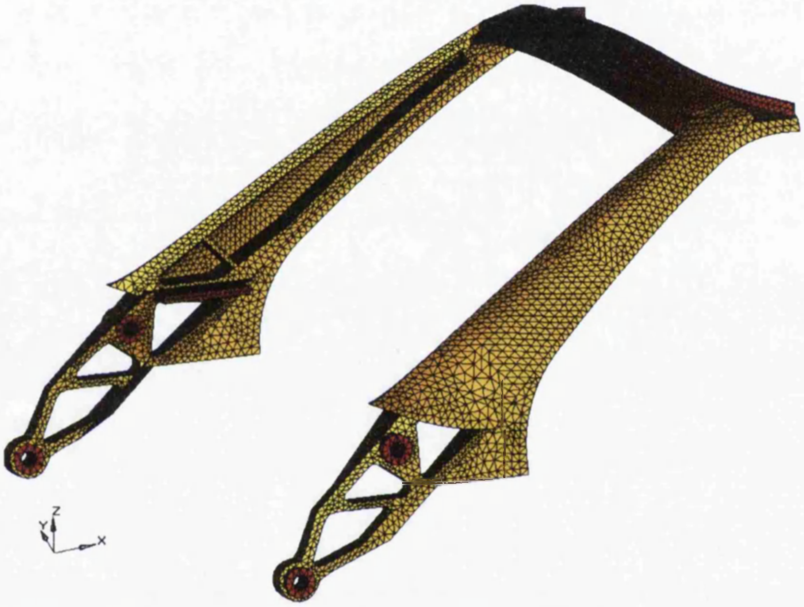


Figure 6.15: FE mesh of the manufacturable design

The mode of structural failure for the A-pillar cast component would be to fracture before the FMVSS displacement limit of 127mm has been reached. Consequently, and from Ref. [6.4], a plastic strain of 10% within the A-pillar was considered as the limit of the design and the yield stress was conservatively assumed as 250MPa. Figure 6.16 shows the simulation results for the revised design. It can be seen that the predicted reaction force is 30kN at the 10% plastic strain limit. Also, the deflection at the 10% plastic strain limit was 90mm and the deflection at the reaction force limit of 17kN (Equation 6.1) was 28.8mm. The A-pillar and windscreen assembly was also assessed for first natural frequency. The first mode was found to be well above the 20Hz target with a value of 51.8Hz. Therefore, the design is feasible with a considerable margin. This indicates that there is further scope for optimizing the design.

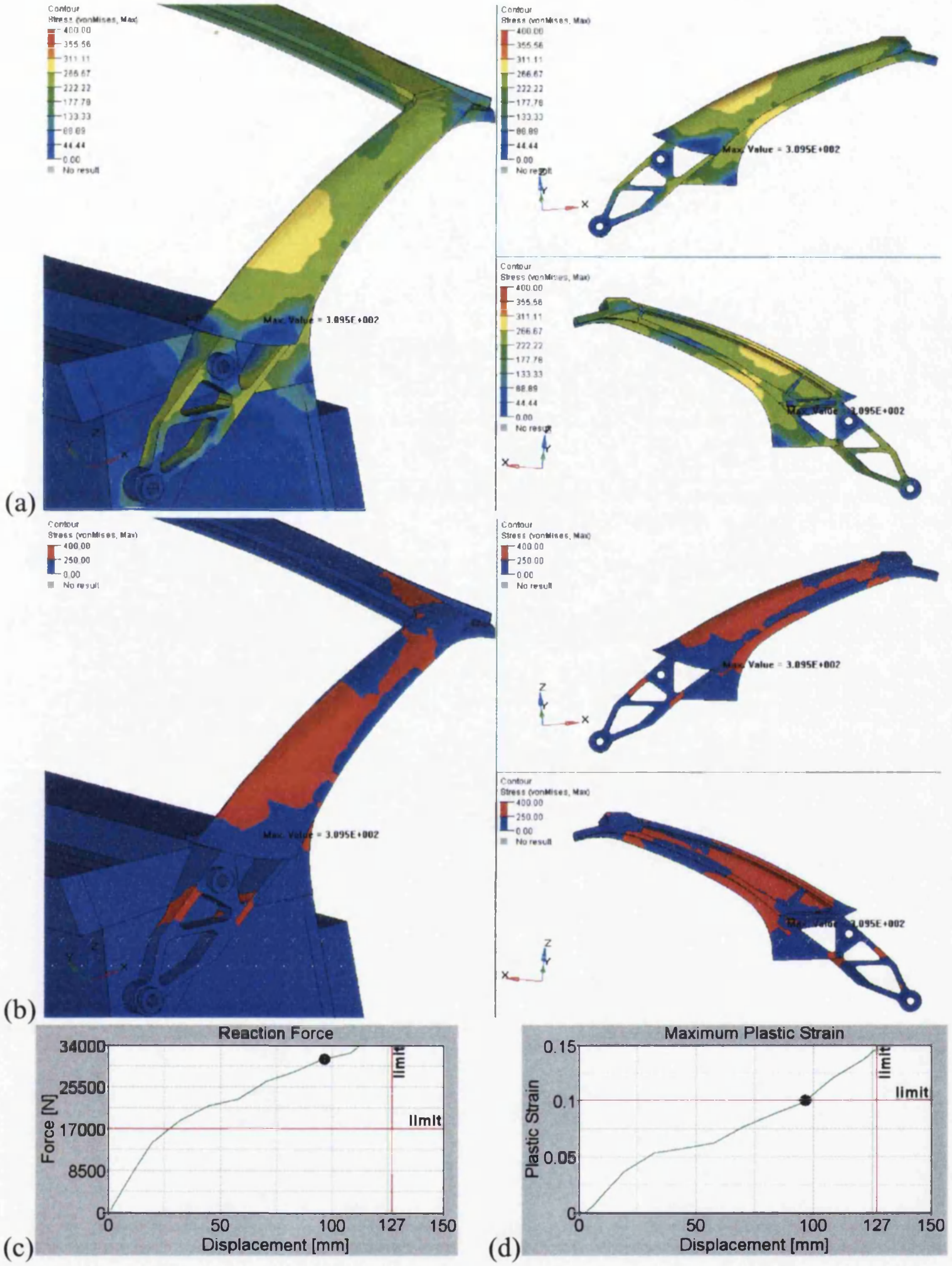


Figure 6.16: Roof crush analysis results, for the baseline design, at a displacement of 90mm. (a) Von Mises stress distribution; (b) plastic strain regions (red) (c) reaction force and (d) plastic strain

6.5.2 SHAPE OPTIMIZATION

The topology optimization undertaken in Section 6.3 resulted in an optimum material layout. The next stage in the process is to use shape optimization to optimize individual parameters such as thicknesses and radii, thereby fine-tuning the performance of the design. The design variable vector consists of three shape variables, parameterized in the FE model using Altair Hypermesh. The shape variables are located in the upper section of the A-pillar where the majority of the mass is located and so will have the greatest impact on the mass reduction. Figure 6.17 shows the variation of the shape variables: shape variable 1 is the thickness of the inner rail that runs up the length of the A-pillar, and varies from 4mm to 15mm; shape variable 2 is a tapered section near to the fixing point with the IP beam, and varies from 4mm to 32mm; shape variable 3 is the outer wall thickness of the A-pillar, which varies from 4 to 8mm. The minimum value of 4mm is used for manufacturing considerations.

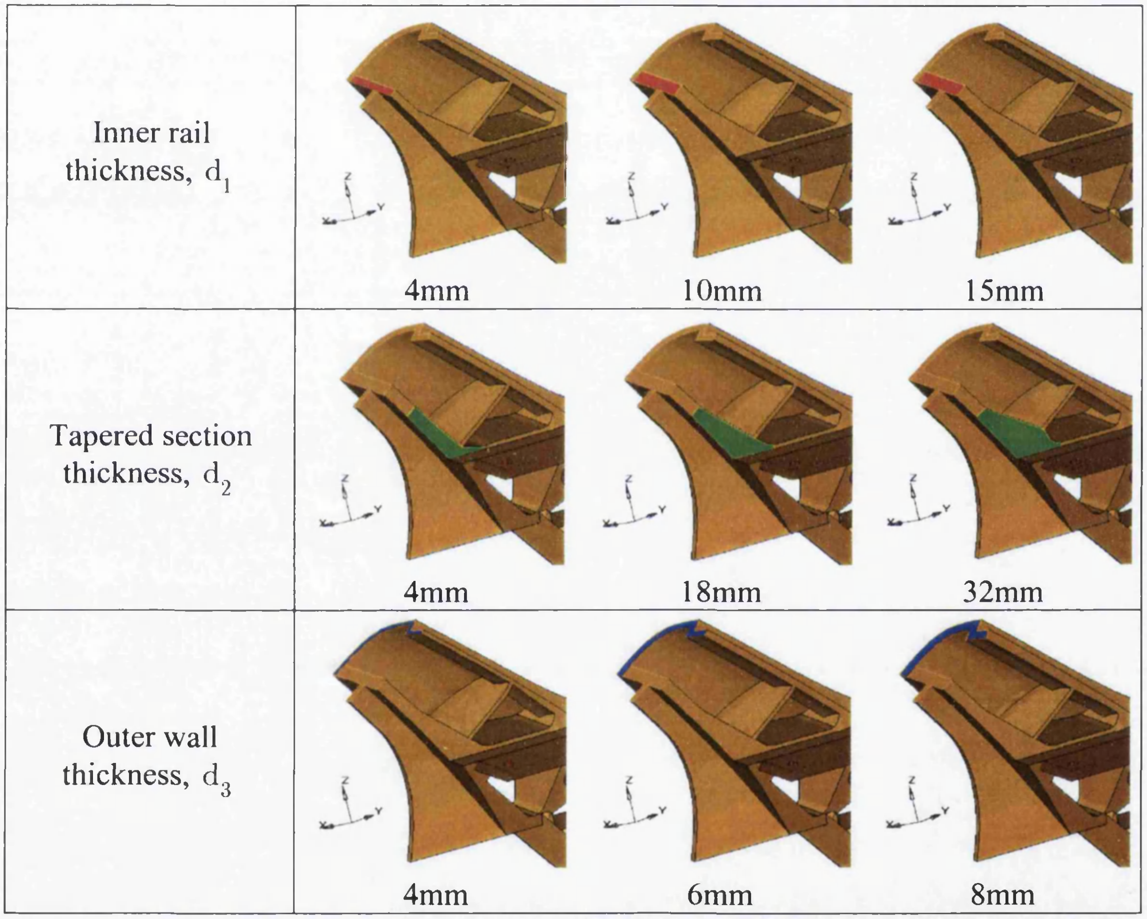

Inner rail thickness, d_1
4mm
10mm
15mm
Tapered section thickness, d_2
4mm
18mm
32mm
Outer wall thickness, d_3
4mm
6mm
8mm

Figure 6.17: Shape variable definition, left to right: minimum, baseline, maximum

There are two main options for carrying out the shape optimization. They differ in the way the performance is calculated. The first option is to run the simulation code, i.e. LS-DYNA, to determine the performance of a design every time the optimizer requires it. The second option is to use response surface modelling [6.10], whereby a DoE is run on the shape variables and an approximate analytical representation of the design space or response surface is generated to replace the computationally expensive simulations in the optimization loop.

The advantage of the first option is that a more accurate prediction of the performance is obtained and so the optimizer is more likely to find the true optimum solution. But the disadvantage is that the computational effort required to reach a solution is greater, it takes approximately 15 minutes to calculate the performance using the simulation code, whereas using the approximate analytical function takes a fraction of a second. Ideally the first option would be used, but in a commercial environment, with the inherent competition for computational time the first option is infeasible.

HyperStudy [6.11] was used to generate second-order approximation functions to replace the analysis code using the Box-Behnken DoE (see Chapter 3 for more details) for three variables. This required 13 LS-DYNA analyses and resulted in Equations 6.3a to 6.3d, which approximate (a) the mass, (b) the plastic strain, (c) the displacement and (d) the reaction force responses respectively.

$$\begin{aligned} F(\mathbf{d}) = 4.65506 &+ 0.24913d_1 - 0.00251d_1^2 \\ &+ 0.04756d_2 + 0.00021d_2^2 \\ &+ 0.29406d_3 - 0.00178d_3^2 \end{aligned} \tag{6.3a}$$

$$\begin{aligned} \varepsilon_p = 0.04908 &+ 0.01981d_1 + 0.01343d_1^2 \\ &+ 0.01072d_2 + 0.00925d_2^2 \\ &- 0.01889d_3 - 0.03518d_3^2 \end{aligned} \tag{6.3b}$$

$$\begin{aligned} u = 26.44027 &- 3.99185d_1 + 4.46508d_1^2 \\ &+ 0.65849d_2 + 0.46754d_2^2 \\ &- 11.84022d_3 + 10.404658d_3^2 \end{aligned} \tag{6.3c}$$

$$P_r = 16995.0 \tag{6.3d}$$

Using these approximation functions, the shape optimization of the baseline design was carried out to solve the following problem, defined mathematically as:

$$\begin{aligned} \text{minimize:}\quad & F(\mathbf{d}) \\ \text{Subject to:}\quad & \varepsilon_p \le 10\% \\ & u \le 127\text{mm} \\ & P_r \ge 16995\text{N} \\ & \omega > 20\text{Hz} \\ \text{Side constraints:}\quad & 4\text{mm} \le d_1 \le 15\text{mm} \\ & 4\text{mm} \le d_2 \le 32\text{mm} \\ & 4\text{mm} \le d_3 \le 8\text{mm} \end{aligned} \tag{6.4}$$

where $F(d)$ is the objective function i.e. mass of the A-pillar for design variable set d, ε_p is the maximum plastic strain, u is the displacement of the test rig, ω is the first natural frequency and P_r is the reaction force of the structure. It should be greater than the required reaction force $P_{r_{req}}$.

For both, the baseline and the optimized models, the design objective, constraints and shape variables are presented in Tables 6.1 and 6.2. It can be seen, that both the baseline and the optimized designs satisfy all of the constraints. For the optimized design, the design variables all reduce in comparison to the baseline design. Variables d_1 and d_3 are at the lower bound of their design ranges, i.e. 4mm, and d_2 takes a value between the upper and lower bounds i.e., 12mm. As a result of this, the mass of the optimized A-pillar is 14% less than the baseline design; overall, a 69% reduction in weight from the intial package space to the optimized model has been achieved. Table 6.2 also gives the percentage difference between the predicted responses from the approximations in Equation 6.3 to the results from the FE analysis. Overall, it can be seen that the approximations give conservative estimations of the responses.

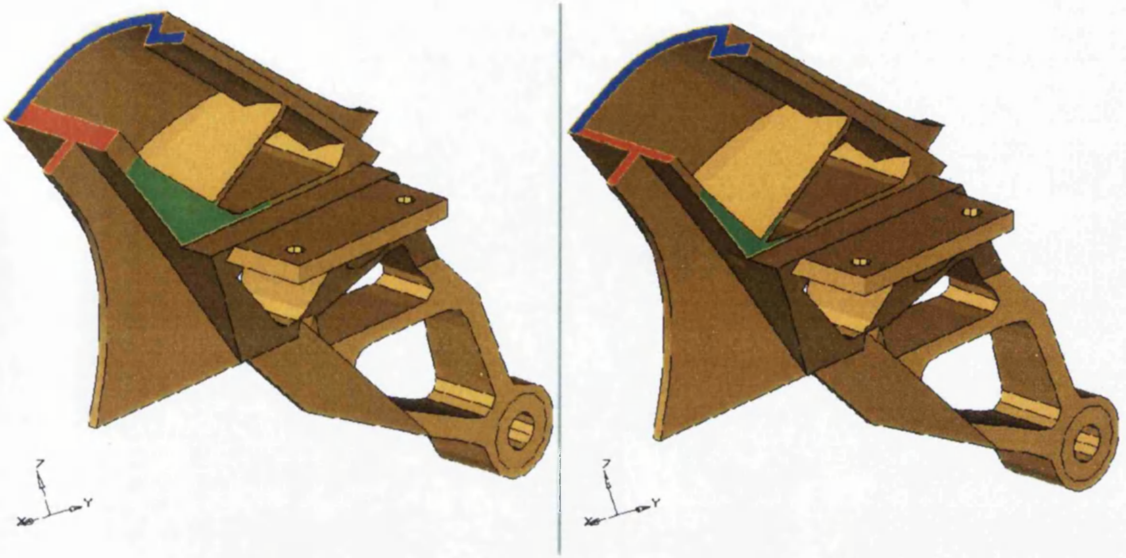
Design variable	Design variable minimum	Design variable maximum	Baseline model	Optimized model
d_1	4.0	15.0	12.0	4.0
d_2	4.0	32.0	27.0	12.0
d_3	4.0	8.0	6.0	4.0

Table 6.1: Shape variable comparison (baseline and optimized) in mm

Objective & Constraints	Baseline	Optimized	Optimized approximation (predicted from Equation 6.3)	% difference between approximation and FE analysis
$F(d)$ [kg]	4.50	3.88	3.92	+1.0
ε_p [%]	5.0	8.5	10	+17.6
u [mm]	28.2	51.0	59.0	+15.7
P_r [N]	16995.0	16995.0	16995.0	0.0
ω [Hz]	51.8	51.1	-	-

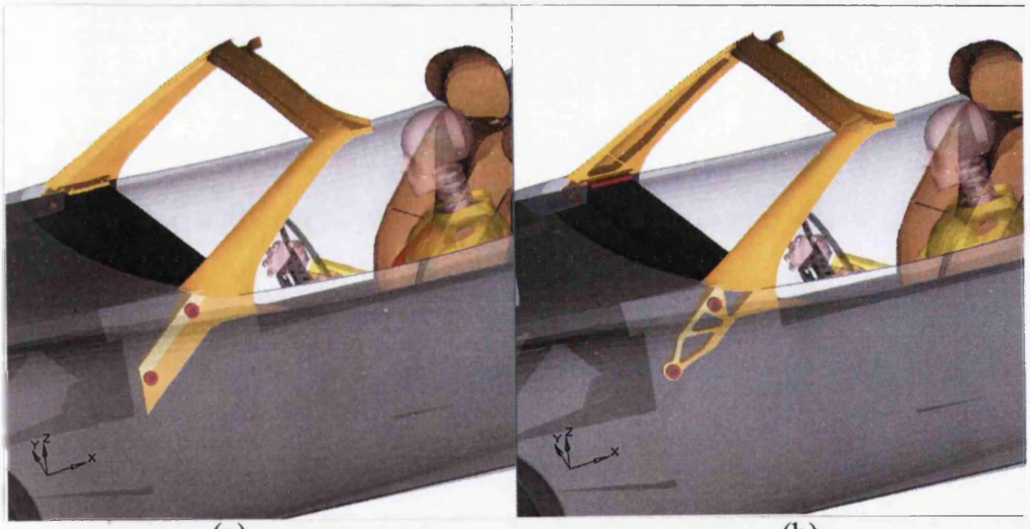
Table 6.2: Objective and constraint comparison (baseline and optimized) and comparison of the responses for the optimized design between the FE analysis and the approximations (Equations 6.3)

The geometric differences between the manufacturable and the optimized design are presented in Figure 6.18 and a virtual visualization of the initial package space and the optimized design is in Figure 6.19.



(a) (b)

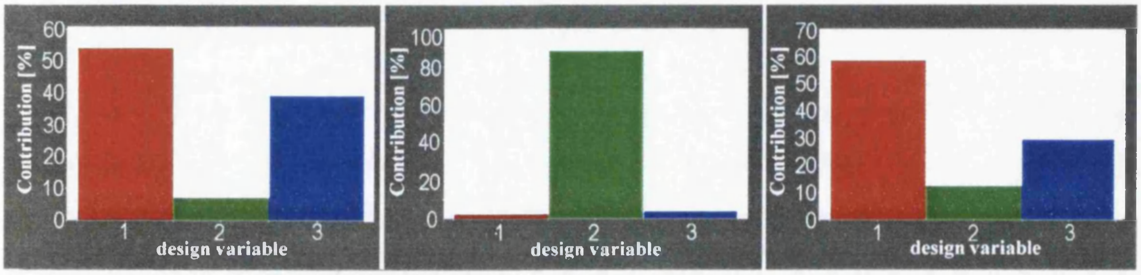
Figure 6.18: Cross-section of (a) the baseline design and (b) optimized design (red = d_1; green = d_2; blue = d_3)



(a) (b)

Figure 6.19: Virtual visualization of (a) the initial package space and (b) the optimized design

The optimization study also provides sensitivity information relating to how design variables affect the objective and constraints. This can be seen in Figure 6.20. It can be seen that d_1 and d_3 have the greatest effect on the displacement and the mass response. d_2 has little effect on the displacement and mass responses, but contributes over 90% to the plastic strain response. These results provide useful information for the designer. Typically, an increased car weight (e.g. different engine) will produce an increase in $P_{r_{req}}$. To cope with this increase, the designer needs to only change d_2, if the plastic strain is undesirable or change d_1 and/or d_3 if the mass/displacement is undesirable.



(a) (b) (c)

Figure 6.20: Sensitivities of (a) the displacement, (b) the maximum plastic strain and (c) the mass of the A-pillar to changes in the shape variables d_1, d_2 and d_3

6.6 Results comparison between the fine mesh and the coarse mesh

This section addresses two questions:

1) What is the effect of using a finer mesh, and hence a smaller minimum member size, on the final material distribution?
2) How does the unaltered topology perform in the roof crush test compared to the baseline solution developed in the preceding sections?

To answer these questions, a comparison between the mesh densities is presented, then, using the finer mesh, two topology results are presented. The first result, uses the same minimum member size as in the coarse mesh topology optimization, the second result, uses a smaller minimum member size based on the fine mesh. Finally, a roof crush analysis using LS-DYNA is carried out on the latter.

6.6.1 Mesh comparison

Figure 6.21 compares the coarse mesh that was used for the concept optimization in Section 6.3, to the fine mesh used in this section. The coarse mesh has 21,221 'designable' elements and the fine mesh has 699,081.

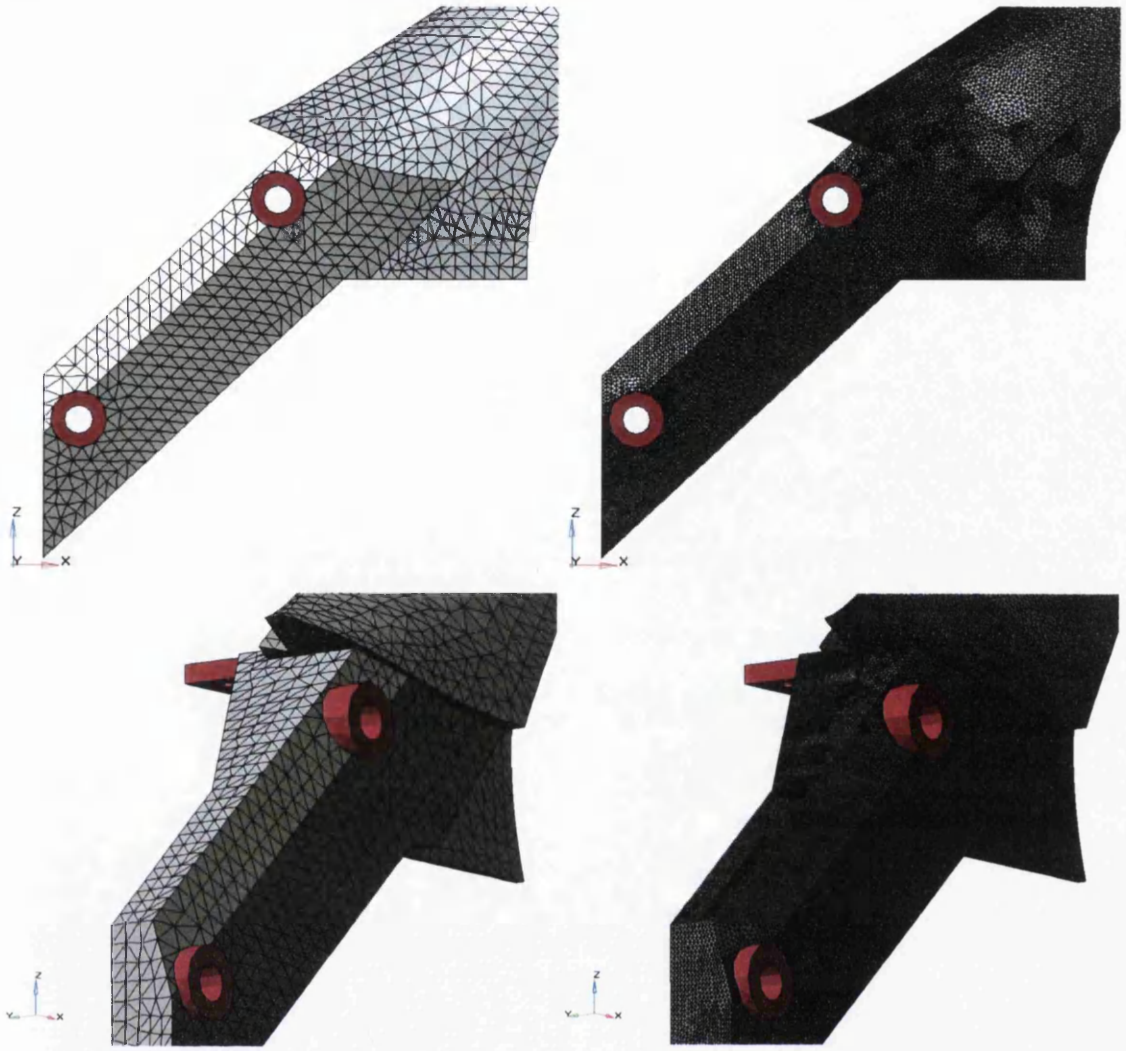


Figure 6.21: Mesh comparison between the coarse mesh (left) used in Section 6.3 and the finer mesh (right) used in Section 6.6; red indicates the non-designable regions where the A-pillar attaches to the body

6.6.2 Topology optimization

Figure 6.22 compares the topology optimization results for the coarse mesh (shown in Figure 6.7(c)) to the result for the fine mesh, both of which use the same minimum member size of 23mm, and are abbreviated as CMR23 and FMR23 respectively. It can be seen that the material distributions are very similar, but the FMR23 result is a great deal 'cleaner'. Both these observations are to be expected. The CPU time to reach the FMR23 was 13 hours and requires 12.6Gb of in-core memory, compared to CMR23, which required 20 minutes of CPU time with 740Mb of in-core memory. Clearly, to generate FMR23 on a PC is currently infeasible due to the lack of memory, which is restricted to 2Gb per processor under a Windows operating system.

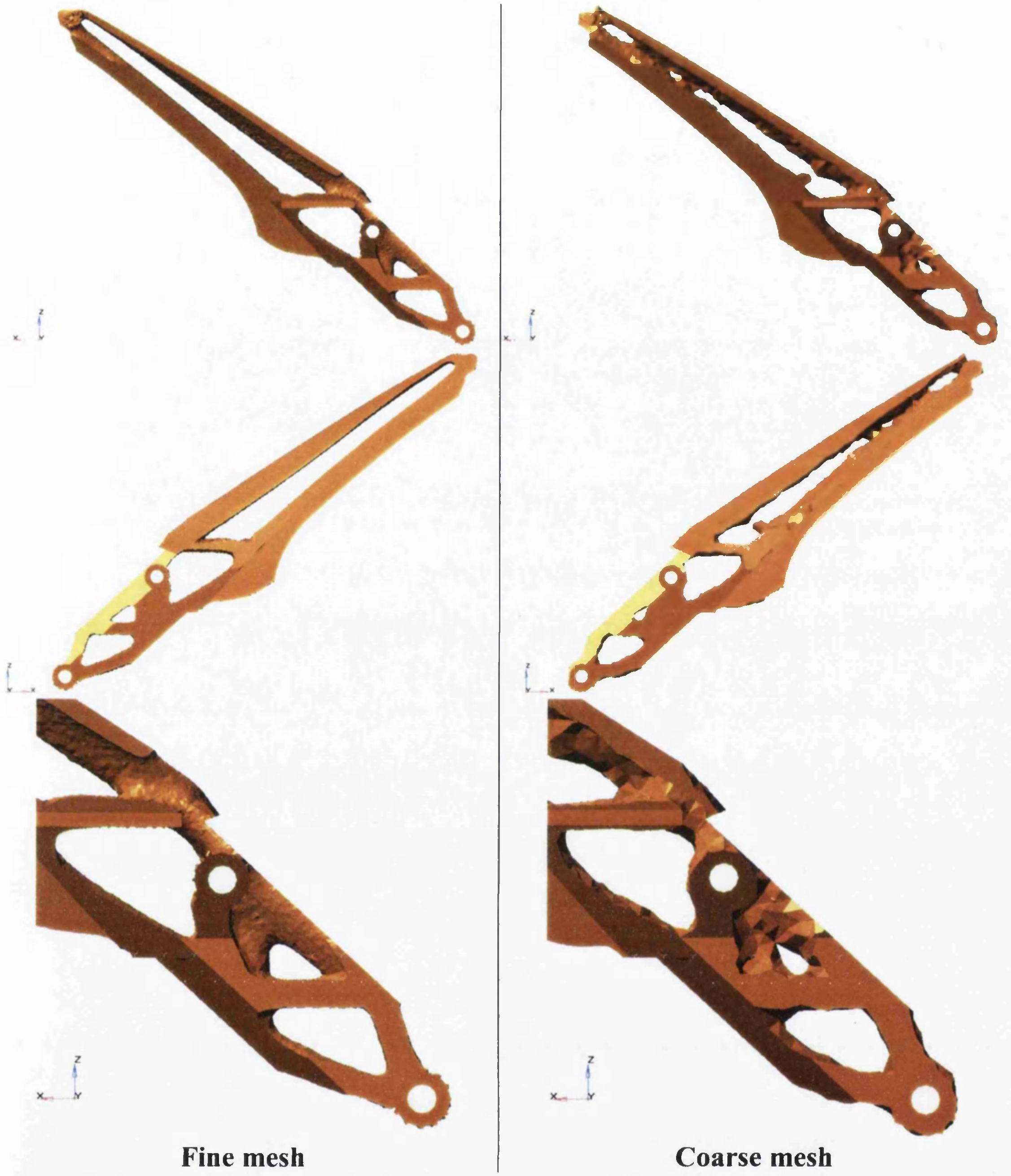


Figure 6.22: Comparison of the topologies for FMR23 and CMR23, both of which use a minimum member size of 23mm

The topology optimization result for the fine mesh, with a minimum member size of 7mm (FMR7), is presented in Figure 6.23. The CPU time to reach convergence for FMR7 is 5½ hours on a 2.6GHz, Pentium 4 PC and requires 5.9Gb of in-core memory. It can be clearly seen from Figure 6.23 that, FMR7 has a cleaner structure than CMR23; and unlike CMR23, there is little ambiguity in the material distribution for FMR7. The use of a smaller minimum member size has produced a different, and more complex, material distribution. Resulting from the use of different minimum member sizes, the following differences between FMR7 and FMR23 can be seen:

- FMR7 has two distinct ribs in the inner rail that runs up the A-pillar.
- The material distributions in the region between the attachment points are distinctly different in their layout; this can also be seen in Figure 6.25.

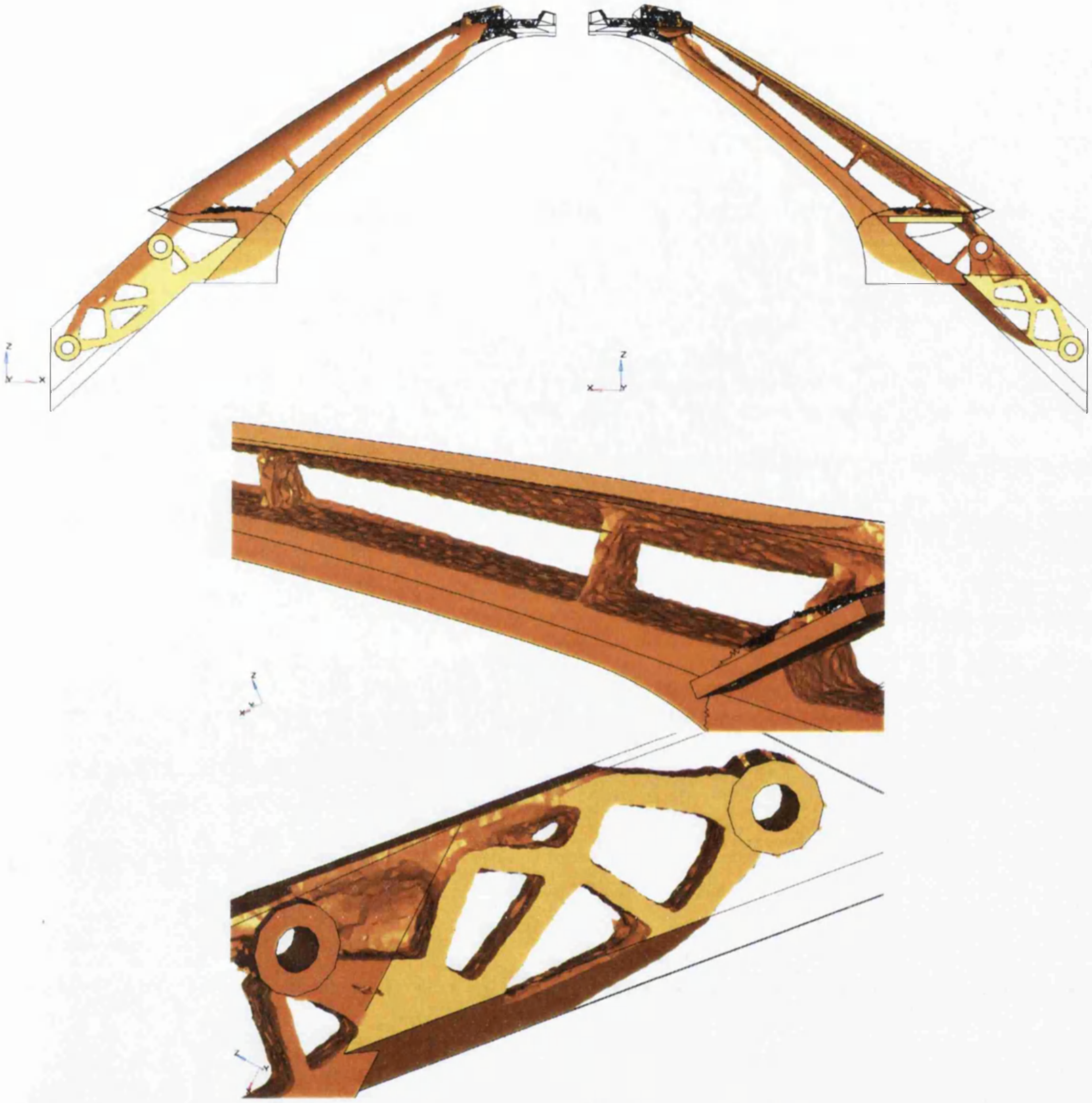


Figure 6.23: FMR7 topology optimization result from various angles for a mass target of 30%, with single and split draw direction constraints

6.6.3 Roof crush test performance

In order to accurately predict the performance of the FMR7 topology, non-linear analysis was done in the same way as described in Section 6.5.1 using the FE mesh of FMR7 as shown in Figure 6.24. The mesh was generated using the geometric definition in the form of an STL file of the final FMR7 topology, and re-meshing it where appropriate. The mass of the resulting structure is the same as the baseline design, i.e. 4.5kg.

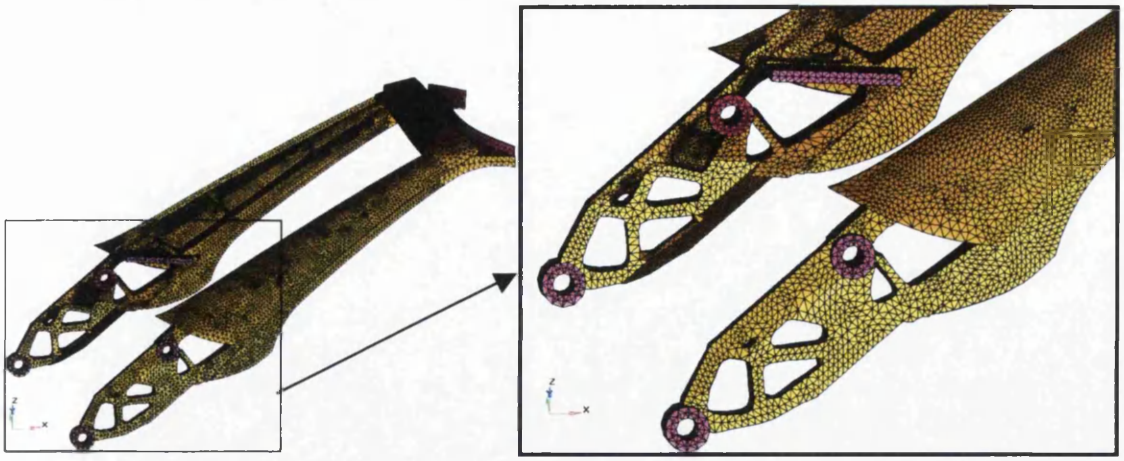


Figure 6.24: FE mesh of FMR7 used in the non-linear analysis of the roof crush performance

A comparison of the structural layout of the baseline design, used in Section 6.5, to FMR7 is shown in Figure 6.25. Figure 6.26 shows the simulation results for the FMR7 structure.

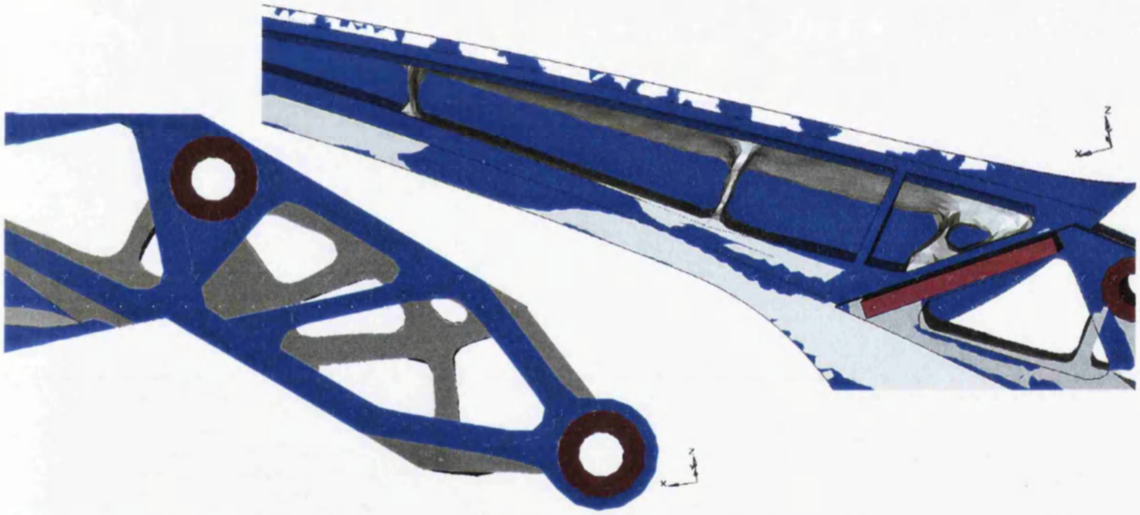


Figure 6.25: Comparison of the A-pillar structures used for the roof crush analysis. The grey structure is from the fine-mesh topology optimization, the blue structure is the baseline design (Figure 6.15)

From Figure 6.26 it can be seen, that the 10% maximum plastic strain constraint is violated for the FMR7 design after a displacement of 35mm. But at this point the reaction force is already 20kN. So, the design still passes the roof crush test. However, it should be noted, that only a few isolated elements have a plastic strain greater than 10% at this displacement, and with some further mesh refinement in this area the performance could be improved. Comparison of the reaction force curves shows that, the result using FMR7 has roughly the same stiffness as the baseline, manufacturable design. The first mode for the FMR7 design is well above the 20Hz target and is the same as the baseline design in Section 6.5, i.e. 51.8Hz.

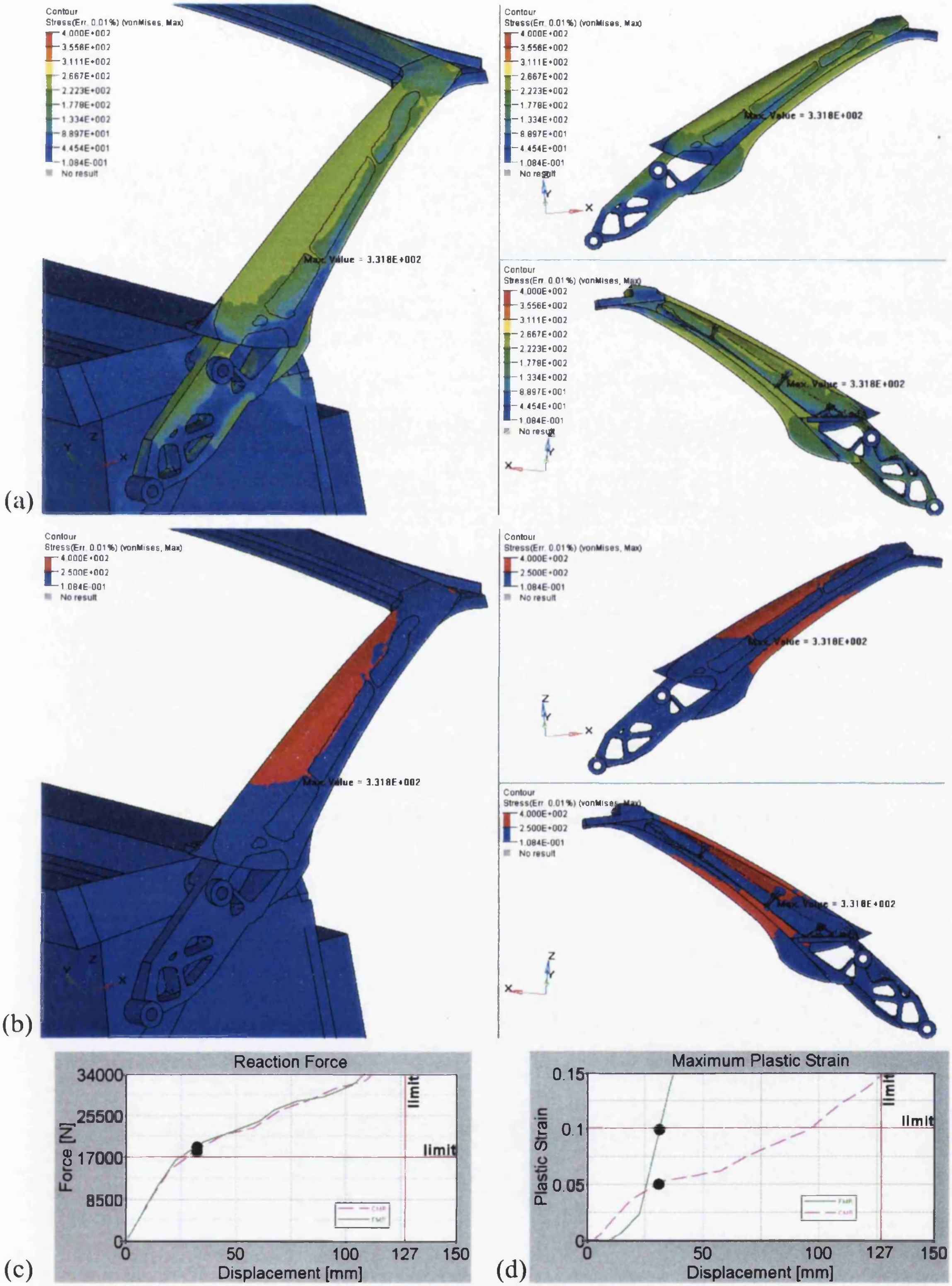


Figure 6.26: Roof crush analysis results at a displacement of 35mm. (a) Von Mises stress distribution; (b) plastic strain regions (red); (c) reaction force and (d) plastic strains for the baseline CMR design (dashed line) and FMR7 (solid line)

6.7 CONCLUDING REMARKS

In this chapter the novel design of an A-pillar has been described. The design meets the FMVSS 216 roof crush requirements. A-pillars are usually designed for extrusion; here the design is for a casting. Manufacturing considerations have been built into the design, and a high confidence level that the design can be successfully cast has been achieved. Through this process, a greater understanding of the physical factors that drive the design has been obtained. The process gives the capability to reach a good design without the need for prior physical prototyping and testing, and thereby reducing time and costs. Typical costs savings are a 20 to 30% decrease in the time to reach a manufacturable solution. The greatest savings, however, come from a reduction in the number of iterations to reach the solutions. Each iteration or refinement requires the manufacture and testing of the component. Using the approach implemented here, a reduction from up to 100 refinements to just one or two refinements can be achieved.

A comparison of the topology results for different mesh densities and different minimum member sizes has been carried out. The results are dominated by the minimum member size constraint, which is enforced when using the manufacturing constraints in OptiStruct. The use of a finer mesh requires considerable additional computer power, but it provides a cleaner design and removes ambiguities in the material distribution. The effect of using a finer mesh, and hence a smaller minimum member size is to generate a more complex and different material distribution. Further work is required to identify the 'optimal' mesh size and minimum member size for which further reduction has no effect on the result or when the resulting material distribution is too complex to manufacture.

The performance of the unaltered fine mesh topology in the roof crush test compared to the baseline solution developed in the preceding sections, shows that the stiffnesses are very similar, meaning that, even after considerable modifications due to manufacturing considerations and engineering interpretation of the CMR results, the baseline design remains very good in comparison to a design which is based purely on FMR.

6.8 REFERENCES

[6.1] Sienz J., "Integrated structural modelling, adaptive analysis and shape optimization", PhD Thesis, University of Wales Swansea, 1994.

[6.2] Campbell J., "Castings"., Butterworth Heinemann, Oxford, 1991.

[6.3] "Federal Motor Vehicle Safety Standards: FMVSS", Sect. 571.216, pp.442-446.

[6.4] Personal communication with Prof. R. Jones, Altair Engineering Limited.

[6.5] "Altair OptiStruct Version 5.1', Altair Engineering Limited, 2002

[6.6] "Alatir HyperMesh Version 5.1', Altair Engineering Limited, 2002.

[6.7] "MagmaSoft", MAGMA GmbH, Germany. info@magmasoft.de

[6.8] Personal communication with Masood Turan, Birmingham University.

[6.9] "LS-DYNA: Non-linear Dynamic Analysis of Structures in Three Dimensions", LSTC, Version 960, 2002.

[6.10] Myers R.H. and Montgomery D.C. "Response surface methodology: Process and Product Optimization Using Designed Experiments", John Wiley & Sons, New York, NY, 1976.

[6.11] "HyperStudy Version 5.0", Altair Engineering Limited, 2001.

CHAPTER 7

SEAT DESIGN FOR COMFORT USING COMPUTER-AIDED ENGINEERING

Summary: This chapter proposes a strategy for designing seats for comfort using computer-aided engineering (CAE), in particular optimization modelling (robust design and topology optimization) and FE simulation. A review of the issues involved with seat design, in particular seat comfort and tissue integrity is also given together with a review of current work on seat design using CAE tools is given.

7.1 INTRODUCTION

This chapter is set out as follows:

Section 7.1: Gives a review of the issues involved with seat design, in particular seat comfort and tissue integrity. Then a review of current work on seat design using CAE tools is given. Finally, important issues and ideas for design directions obtained from the review are consolidated.

Section 7.2: A general procedure to investigate seat design for comfort using CAE is proposed.

Section 7.3: A variety of preliminary studies are carried out:

(i) FE model of a hybrid III dummy sitting on a car seat (courtesy of Jaguar Cars Ltd.) and the corresponding prediction of the interface pressure distribution.

(ii) Pressure distribution mapping of a person seating on a rigid surface contoured to the occupants shape.

(iii) Production of a plaster cast mould of the same person sitting down and the generation of an FE model of the casting using a three-dimensional scanning system.

Section 7.4: Gives some concluding remarks.

Section 7.5: Lists the references used in the chapter.

This chapter is concerned with the development of a strategy for designing seats for comfort using computer-aided engineering (CAE), in particular optimization modelling (robust design and topology optimization) and simulation. These correspond to the second and third columns of the three-column concept depicted in Figure 1.3.

Improving seat comfort is very important in several areas including the automotive and airline industries, where competition is very high, and also in wheelchair design. Quantification of seat comfort is an important requirement, however, it is very difficult to determine. Questionnaires are often the only way of generating quantitative data, as such the information is subjective and in the case of some wheelchair users, who cannot communicate effectively the level of comfort, may be impossible to ascertain. The pressure distribution between the user and the seat is likely to have a major influence on comfort. A typical method for designing seats for comfort would be to use an iterative physical process whereby pressure mats are used to determine the pressure distribution between the occupant and the seat and then the shape and density of the seat foam is altered to improve the uniformity of the pressure distribution. This process is cumbersome and in a highly competitive market, such as the automotive industry, is becoming less and less feasible. Using CAE optimization to model the pressure distribution, the shape and density of the foam can be 'virtually' optimized to maximize the pressure uniformity. The use of CAE is seen as an important and necessary tool, as many more shapes and sizes of occupants can be modelled, and virtual assessment can be made rapidly thereby providing a very good initial design for the physical design process and reducing the number of expensive physical iterations. This also applies to wheelchair seat design, where the occupant can be modelled and the appropriate shape and density of the seat can be determined without the need for much initial communication with the occupant. In Section 7.3 a procedure for developing a CAE optimization model for seat design is proposed and some preliminary work is carried out. This involves using a finite element model of a hybrid III dummy with the model of the Jaguar X350 car seat. Several model set-ups are used and pressure distributions are identified.

Seats are used on a daily basis by almost everybody and their design is very important. They need to be designed for a wide range of occupants, ranging from able-bodied persons to highly disabled wheelchair users, and need to be functional and comfortable. There are a number of issues related to seat design. Tissue integrity management is essential, especially for wheelchair users at risk from pressure ulcers. It is also concerned with the modelling of body tissues, methods for maintaining tissue integrity and the mechanisms behind pressure ulcer formation. Seats need to be designed to minimize discomfort thereby allowing the occupants to maximize their ability to operate normally. There are many types of materials used for seat cushions such as "contoured foam, water filled, solid gel, viscoelastic foam and segmented foam."[7.1] According to Schmeler *et al.* [7.2] the choice of cushion used, especially for wheelchair users, must also consider nutritional status, continence, and co-morbidities, weight, postural stability, reliability, and maintenance.

The work here is concerned with seat design for comfort and tissue integrity. Other issues include the structural design of the seats to withstand daily use and crash impacts.

7.1.1 SEAT COMFORT AND TISSUE INTEGRITY ISSUES

There is no general agreement on the meaning of comfort and discomfort. Some subjective factors linked to discomfort are: "sore muscles, heavy legs, uneven pressure, stiffness, restlessness, fatigue, and pain; and to comfort are: relaxation, refreshed feelings, spaciousness of the chair, liking the chair, aesthetic appearance of the chair, feeling good, feeling supported in the right places, feeling little pressure under the buttocks, feeling stable, feeling satisfied, and several others." [7.3, 7.4]

Seating discomfort, may be defined as: "A negative feeling, reaction or sensation, that usually occurs over time, that can often limit a person's ability to function in their mobility system and therefore their expected or desired role within society. It often first presents itself as an unconscious desire to change body posture. It is often associated with one or more factors such as: sitting instability, forward sliding, excessive heat build-up, stiffness, excessive localized soreness or pain, spasticity, or stretch. It may be specific in location or generalized, but diminishes when the person is able to initiate frequent changes of their seated posture or is no longer in the mobility device. It can be a precursor to the development of secondary conditions such as joint deterioration, ulcer formation, and circulatory disorders." [7.3]

There are two principal categories into which seat users fit. The first is able-bodied persons for which "manufacturers of office, automotive, and truck seats have done extensive product development to enhance seat comfort and user productivity. All of these innovations are based on the premise that normal seated comfort is not derived from a single static posture, but requires changes in posture (dynamic seating) over time and able bodied persons are able to maintain discomfort at tolerable levels through small, unconscious body movements or postural adjustments that maintain discomfort at tolerable levels." [7.3] The second category is disabled wheelchair users. For this population, especially for those "persons with advanced stages of Multiple Sclerosis, Muscular Dystrophy, Amyotrophic Lateral Sclerosis, and some people with Post Polio Syndrome, the discomfort and pain of daily wheelchair sitting can be a chronic problem. Due to their neuromuscular disorder, they have a lack of sufficient motor function to attain adequate redistribution of supporting forces to relieve discomfort. In addition to the potential for pressure ulcer formation, many people in this population experience intolerable periods of discomfort, which can lead to reduced participation in daily activities including work, education and recreation."[7.3]

"Current wheelchair technology is designed mainly for pressure relief for high-level spinal cord injured persons and does not adequately meet the needs of the wheelchair users." [7.3] "In developing countries, pressure ulcers are the number one cause of death among people with disabilities. The resources to purchase or manufacture state-of-the-art cushions are not available. The majority of commercial seat cushions cannot be produced in these areas because they require materials or manufacturing techniques that are not obtainable. Currently, there are very few low-cost seat cushion designs for people with disabilities."[7.5]

For further information on the current issues regarding seat comfort, especially for wheelchair design, the reader is referred to the work of Hobson *et al.* [7.3]

As discussed above, maintaining tissue integrity and the need to understand the mechanisms in pressure ulcer formation is important. For current issues and the state-of-the-art of tissue integrity management, the reader is referred to the work of Brienza *et al.* [7.6]. The main factors involved in the formation of pressure ulcers are as follows (note that the indicated references are taken from Ref. [7.7]):

- Insufficient vascularization in the tissues subjected to high pressure (mainly under bony prominences) due to the occlusion of capillary flow [7.8, 7.9] from interface pressures of 1.5kPa to 16kPa [7.6].
- The stagnation of sweat on the skin as a result of inadequate air replacement. [7.10]
- The presence of local areas of high temperature. [7.9, 7.11]
- Shear stresses on the skin. [7.12]
- Studies show that interface pressure is the principal factor involved in the formation of pressure sores. It is widely accepted [7.9, 7.13, 7.14, 7.15] that the least possible pressure be placed on the tissues.

Pressure management is therefore important and Ref. [7.1] gives recommendations to: maximize the surface area to decrease the pressure on any one location (peak pressures); redistribute body weight using support surface shape and materials with required properties; minimize asymmetries i.e., unequal loading of pelvic structures and tissues.

Furthermore, as referenced by Brienza *et al.* [7.6], Levine *et al.* [7.16] state that tissue integrity is maintained by reducing pressures near bony prominences, accommodating orthopedic deformities through immersion, enveloping irregularities at the seating interface to reduce high-pressure gradients, and controlling heat and moisture.

In Ref. [7.7] interface pressure between the buttocks and the seat cushion was used to provide peak pressure values and information on the contact surface and the postures associated with four different pressure-relieving cushions. Combined with clinical judgment, pressure-mapping information was shown to have some predictive ability to determine risk for the development of pressure sores. In Ref. [7.17, 7.18] studies into the efficacy of pressure reducing cushions for high-risk elderly people who use wheelchairs showed that people with higher interface pressure measurements had higher associated incidences of sitting-acquired pressure ulcers. In Ref. [7.19] a similar study was conducted, which found a significantly higher incidence of pressure sore development in people with high peak pressure measurements as compared to those with lower peak pressures [7.2]. In [7.20] the authors describe pressure mapping as being clinically useful for comparing cushion effectiveness, and understanding what may be causing pressure sore problems. However, care should be taken in using interface pressure as the criterion for good seating since its use "has considerable limitations"[7.21, 7.9] and "pressure alone is not necessarily a predictor of a person's risk for the development of a pressure sore." [7.2]

According to Levine *et al.* [7.16] (referenced by Brienza *et al.* [7.6]) tissue deformations are influenced by the magnitude, direction and the distribution of forces over the body. Typically, the loading conditions are a combination of normal and shear forces transmitted to the tissue causing deformation, resulting in internal stress and strain of the tissue that can lead to necrosis. Internal tissue pressure levels are increased by tissue deformation and the increases depend on the mechanical properties of the tissues and the directions in which they distend. If the tissues are confined so that no redistribution of tissue mass can occur or if the loading is applied hydrostatically, the soft tissues can withstand relatively high pressures (typically 10-27kPa) without significant risk of tissue damage. Only when pressure is applied non-uniformly are tissues strained and consequently put at risk of tissue damage.

According to Karg *et al.* [7.21] the optimum seating will be achieved when the soft tissues are subjected to the least distortion. They say, however, that meaningful measurements of tissue distortion, tissue thickness, shear stress, and tensile and compressive forces have been difficult to obtain and that significant stresses occur within the subcutaneous tissues due to distortion of the tissue from seated loading. In theory, the optimum shape is "the one that would minimize the internal tissue stresses and external pressure on the tissue."

In order to relieve pressure, mechanisms have been introduced to alter the distribution of buttock loading by alteration of seated posture using tilt-in-space and back recline wheelchair seats, the problem with this is that the mechanism most often places a person in a non-functional position thereby defeating the object of designing for seat comfort.

7.1.2 CURRENT RESEARCH IN DESIGNING SEATS FOR COMFORT

Much of the research into seat design for comfort is being pioneered by the automotive industry. This is mainly due to the fact that the industry is highly competitive and as such attention to detail is essential. However, very little of this research has been "routinely applied to wheelchair seating"[7.3]. In general, using CAE helps to improve efficiency and to reduce costs. For example, in the automotive industry where products are constantly being modified and need to remain cost effective, CAE can help to assess many possible solutions in minutes rather than weeks. For instance, Jaguar Cars Ltd. [7.22] currently uses questionnaires (with specially trained comfort assessors) and physical pressure mapping to identify regions of discomfort. They then modify the seat foam to improve comfort. This process is repeated several times, and it is time consuming and cumbersome. The use of CAE with, for example, a computer model of the pressure distribution correlated to questionnaire results is seen as an important area of research because it can give a good starting point for a design and cut down the number of iterations required to reach a final, satisfactory design. In this section references are made to current research using CAE for seat design.

As stated earlier "optimum seating will be achieved when the soft tissues are subjected to the least distortion." Brienza *et al.* [7.23] have attempted to use this criterion in a semi-computational manner. Here optimization is carried out to find the seating surface shape that minimizes variation of the buttock soft tissues from their unloaded shape, indicated by the tissue stiffness. The "simulation" in the optimization loop is not computational, but is carried out using a physical system called 'Computer Automated Seating System' or CASS, shown in Figure 7.1. The system controls the seating surface shape while measuring the external pressure applied to the buttocks by the surface. The results show that lower peak interface pressures are found with the optimal support shapes than are obtained with the flat foam. CASS was improved in Ref. [7.24] with the ability to directly measure the thickness of multiple soft tissue layers using ultrasound pulses instead of inferring the tissue thickness from the measured stiffness, as was done previously.

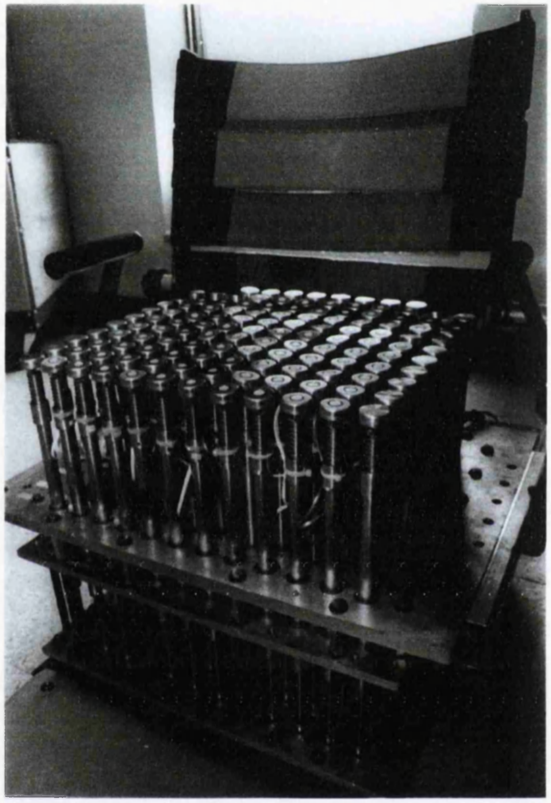


Figure 7.1: Latest version of the Computer Automated Seating System (CASS) developed in [7.24]

To exploit the full advantages of CAE, the system should be fully computational and an appropriate model of the human body, usually finite-element (FE) based, is needed to be able to compute the expected behaviour. Ideally a "complete, perfect FE model of the human body includes all tissues that possibly participate in the continuum mechanical behaviour of the body, such as skin, muscles, tendons, ligaments and fat, and it describes the morphological properties. In addition each tissue is defined by geometry, material properties (anisotropic, non-linear, time-dependent), and contact conditions." [7.25]

Moes *et al.* [7.26] started to develop a CAE system by identifying how the pressure distribution is related to comfort in terms of physiological and biomechanical factors. In Ref. [7.27], they measured the pressure distribution for subjects sitting on a flat, hard and horizontal support and the relationships between the pressure distribution parameters, the body characteristics and the angle of the pelvis rotation were identified. A method for developing a geometrical model using a three-dimensional scanner followed by the subsequent development of an FE model was carried out in Ref. [7.28], where further information can be found on various geometry capturing techniques including: using slices, MRI scans, CT scans, surface scanning, ultrasound and contour mapping. This was extended in Ref. [7.29,7.30] with the development of a generic model for the shape of a part of the human body, namely the buttocks and upper leg area. The model is called 'vague' and is derived to reflect the uncertainty in the human shape.

In Ref. [7.31] an electronic measurement system called PCMAN has been developed to produce three-dimensional computational models of the human in static and dynamic postures. PCMAN works with several images taken from different directions with standard cameras. From these, three-dimensional coordinates can be calculated by measuring corresponding points on the video frames.

In order to have a valid model, estimation of the elastic properties of the skin is required. In Ref. [7.25, 7.32, 7.33, 7.34] the material properties of the FE model described by Moes *et al.* [7.29, 7.30] are determined using optimization, with the aim of matching the pressure distribution predicted by the FE model to the real pressure distributions found in Ref. [7.27]. Other investigations into the mechanical properties of the human skin were carried out in Ref. [7.35] as referred to in [7.25]. In Ref. [7.25] Moes *et al.* developed an FE model (a simplification of the ideal model) of the human body that consists of skin, bone and soft tissue. They then applied optimization to improve the ergonomic quality of the initial contact shape of a product by maximizing the *Ergonomics Goodness index*, ϵ, introduced in Ref. [7.36]. The index relates product shape to the pressure distribution on the FE model and uses a weighted sum of several factors including: physiologically acceptable pressure, the curvature of the surface, sudden protrusions from the surface called 'sharp singularities', sudden changes in the curvature of the surface called 'phantom singularities' and the average pressure.

Ergonomic factors have been used in a number of other studies to improve seat comfort and, as stated in [7.37], seating ergonomics is dominated by subjective investigations [7.38] or investigations based on questionnaire surveys [7.39] and FE-based CAE is being increasingly used. It is evident from the questionnaire results [7.22] of occupants assessing seat comfort of car seats that if the same seat is put into two different cars then the occupant's perception of comfort will be different. This means that comfort is not necessarily to do the with pressure distribution on the seat but also to do with the surroundings. Kolich [7.40] warns that "automobile seat comfort is a unique science. Ergonomics criteria, while serving as the basis for this science, cannot be applied blindly for they do not ensure comfortable automobile seats."

Another consideration of computer modelling is being able to model human posture. RAMSIS (Computer-based anthropological-mathematical system for passenger simulation) is an ergonomics CAD tool developed in the research presented in Ref. [7.41]. It is used to test the arrangement of components within, for example, automotive interiors by predicting occupant postures. The PCMAN model described earlier was developed with the aim of interfacing with RAMSIS in order to facilitate rapid assessment of different body shapes and sizes.

In [7.42] a rigid body model of the human body sitting on a typical office chair is used to investigate the effects of changing seat tilt, backrest recline, backrest profile and backrest height on the curvature of the lumbar spine in the sagittal view (side view of the human body).

Stelze and Johnson controls (JC) [7.43] use a method called VIPS (Vibration insulation, Pressure distribution, and H-point simulation) to get the quantitative information used in the seat comfort optimization. Here, the pressure distribution is calculated depending on the material of the seat upholstery and the digital dummy used, and the higher pressure points are assumed to indicate regions of discomfort. Currently, JC are developing Cosyman [7.44] - Comfort Optimization System with Mechanical Analysis, with the aim being to include the deformability of the spine in the simulation.

In response to the fact that "static sitting can constrict circulation over time" a system called 'ComfortRenews' was introduced by Johnson controls [7.45] to alleviate restlessness and fatigue by creating a natural walking-like motion required for proper circulation. Figure 7.2, taken from Ref. [7.45] shows the idea behind the mechanism.

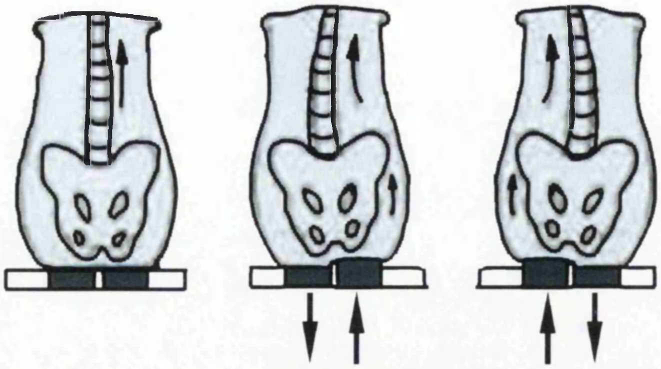


Figure 7.2: Schematic view of the 'walking-like' motion invoked by a seat using the 'Comfort Renews' device developed by Johnson controls [7.45]

According to Ref [7.46], Swiss seat supplier Lantal Textiles has developed an inflatable aircraft seat which it claims is more comfortable and 5.9kg lighter than normal seats. They have substituted air cushions for the usual rubber foam seat material, and passengers can pump air around the seat to achieve the most comfortable shape. Johnson controls take away the occupants input in their system called 'ComfortConforms' [7.45]. The method changes the contour of the cushion using air cells (positioned on top of the seat foam) to fit the occupant by automatically measuring the occupant's weight distribution and providing support accordingly. The "comfort algorithm" used to provide the appropriate support is based on data gathered from consumer testing and aims to eliminate pressure points.

Current research into seat comfort is based dominantly on pressure re-distribution. However, Pankoke *et al.* [7.47] assessed comfort according to the level of vibration. In that work a CASIMIR FE model of the human body is used to compute mechanical quantities such as deformations, accelerations and forces of the body and inside the body. All of which are considered to be essential for an objective and reproducible valuation of vibration comfort. The model is interfaced with RAMSIS and is used to evaluate vibration comfort for a broad variety of individuals, represented by age, nationality, gender and anthropometric status.

Bartels [7.48] studies the influence of cover and cushion materials on the thermal comfort of aeroplane seats. Seat trials with human test subjects were performed in a climatic chamber. It was shown that the physiological seat comfort of aeroplane seats could be considerably improved by using textile covering layers and cushions. Fabric has better moisture transport properties than leather, both for normal seat situations and heavier sweating. It was also concluded that a three-dimensional knitted spacer fabric is a better design solution than a moulded foam pad.

Huizenga *et al.* [7.49] have developed the 'Berkeley Comfort Model' to predict human physiological response to transient, non-uniform thermal environments. The model allows unlimited body segments, each modelled as four body layers (core, muscle, fat, and skin tissues) and a clothing layer. Physiological mechanisms such as vasodilation, vasoconstriction, sweating, and metabolic heat production are explicitly considered. Convection, conduction (such as to a car seat or other surface in contact with any part of the body) and radiation between the body and the environment are treated independently.

Rasmussen *et al.* [7.37] have developed a musculoskeletal model that can calculate how muscular activity is affected by changes in sitting conditions. A detailed human body calculation is made and with the model it will be possible to examine different postures and the associated support forces in order to minimize discomfort. An example of the seated human project (Rasmussen *et al.* [7.37]) is shown in Figure 7.3. It comprises 84 muscles, 16 segments, and 9 joints.

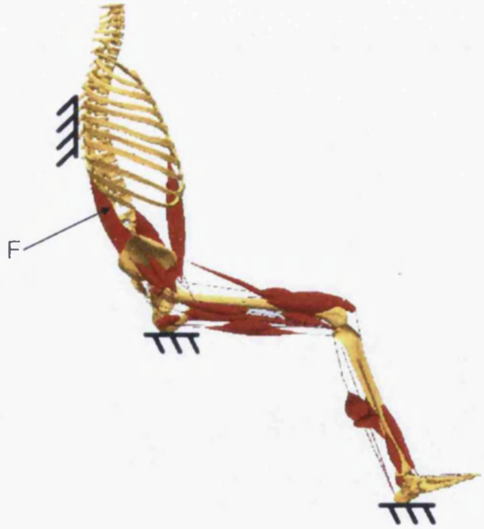


Figure 7.3: A musculoskeletal model of a seated human in the AnyBody system. The variable force F represents the lumbar support. (Rasmussen *et al.* [7.37])

Here the optimization of lumbar support for different backrest inclinations was considered. The lumbar support is represented in the model by a variable force perpendicular to a lumbar segment. By using the "AnyBody" modelling system for calculating the corresponding muscular activity for different magnitudes of F, the magnitude of the lumbar support force that enables the minimum activity of all muscles is identified. This is considered optimal for comfort.

7.1.3 CONSOLIDATION

Much of the research is based on finding the shape of the support surface. There is stochastic variation in the shape and the size of the human body. Therefore an approach such as robust design is essential to designing practical seats. In order to address the need for cheap easy to manufacture seating for wheelchair users the seats need to be simple, use readily available materials and be cheap. The methodology for designing the seat has to be adapted such that it can be easily used to produce designs specific to individual, users such as severely deformed disabled users, but also be used to design seats for the mass market.

Most of the literature looks at shape optimization to get the contour of the seat cushion based on ergonomic considerations and pressure distributions. If for instance, the designer fixes the shape of the cushion, then alteration of the pressure distribution is achieved by alteration of the angles of the support surfaces or altering the internal layout of the support surfaces. The latter is usually achieved by an expensive mechanism such as using air-filled cells, which is infeasible for wheelchair seating in developing countries. No information has been found on the CAE design of the internal material layout of cushions. Topology optimization is a tool naturally suited to this problem and the development of such a tool to find the optimal distribution of foam in cushions is likely to achieve greater comfort for the occupant than is possible with a standard uniformly distributed foam, and at very little extra cost.

A promising area of research is the AnyBody system being developed by Rasmussen *et al.* [7.37]. Using this research to find the orientations of the support surfaces coupled with the use of topology optimization to find the internal layout of the support surfaces could produce designs that will minimize the need to alter the orientation of the seat and also minimize the need for the user to have to move in the seat to relieve pressure.

There are many factors that influence comfort of which pressure distribution is considered a major factor. Quantification of seat comfort is extremely difficult due its subjectivity. A necessary assumption to begin to look into the seat comfort quantification is that the more uniform the pressure distribution is, the more comfortable the seat.

As far as developing a CAE model is concerned it is apparent from the literature that the starting point of much of the research is complex. Starting from a complex shape such as the human body is likely to be too advanced when starting to develop a CAE model and a simpler starting point is recommended. For example, the starting point could be to model a cube of homogeneous material placed on a hard surface and the FE results of the pressure distribution can be easily correlated to those from a pressure mat when compared to the complex human form.

7.2 General procedure

The overall objective of the proposed procedure is to develop a computer based system for designing comfortable seats that can be adapted to all shape and sizes of occupant. It is assumed that reducing peak pressures and improving the uniformity of the pressure distribution maximizes comfort. In order to achieve this objective it is necessary to break the process down. The first simplification is to develop the system using rigid-surface objects such as a rigid dummy. This offers three key advantages:

(i) Using rigid-surface models removes the need to model the complexities of the human body such as the tissue deformation.

(i) Computer crash test dummy models that have been correlated to their physical counterparts are available so that physical pressure mapping tests can be carried out virtually and physically without much discrepancy between the results.

(ii) There is no need to create complex FE models of different people using for example, three-dimensional scanning.

Once the system for designing the seats is developed for rigid-surfaced models, then the same system can be applied to a more realistic model using an independently developed complex FE model of the human body. Note, the orientation of the support surfaces could be determined *a-priori* using a method such as 'AnyBody' by Rasmussen *et al.* [7.37], to minimize muscular activity. This ensures that a degree of comfort will be automatically factored into the initial seat design.

The following steps indicate a general procedure to develop a CAE model for use in predicting pressure distributions and optimizing seats for comfort. Where appropriate a table summarizing the input, the work and the output for a step is given.

Step 1

Model several homogeneous objects of varying complexity placed on a hard surface and correlate the FE results of the pressure distribution to those from a pressure mapping system.

INPUT	WORK	OUTPUT
Several homogenous objects with varying shape complexity ranging from a cube and finishing with a dummy.	Physical pressure mapping of the objects on a rigid surface. FE models of the objects, followed by FE analysis to determine virtual pressure distributions. Followed by correlation of the FE model to the physical model.	FE model for prediction of pressure distribution for several homogenous objects with increasing complexity on a rigid surface.

Step 2

Using the model developed in Step 1, replace the hard surface used with varying supporting surfaces and mechanisms. These may range from a foam layer on a hard surface to a full FE model of a car seat.

INPUT	WORK	OUTPUT
The FE model generated in Step 1.	Replace the hard surface with different models, starting from a model of a foam layer supported on a rigid surface, ending up with a complex model such as a car seat.	FE model for prediction of pressure distribution for several homogenous objects with increasing complexity being supported on surfaces of increasing complexity.

Step 3

In order to carry out optimization an objective function is required, so the purpose of this step is to define an objective function to quantify comfort. There are two possible options. Firstly, it could be assumed that a uniform pressure distribution is equivalent to the most comfort. Thus, an objective function such as the uniformity of pressure could be used. Or secondly, an objective function could be derived through correlation of questionnaire and pressure mapping data. Jaguar Cars Ltd. has carried out extensive comfort studies using occupants specifically trained to assess comfort and have a great deal of data available to facilitate this second option.

Step 4

The aim of this step is to identify the key controllable and uncontrollable system parameters and their variations, then to carry out DoE studies to identify the parameters with the most influence on the objective function defined in Step 3. This step also involves parameterizing the seat cushion to define a set of design variables. These variables can be shape variables used to alter the outer surface of the cushion or variables that alter the internal material density distribution for use in topology optimization. The latter option is more appealing since no research into this area has been attempted previously and the shape of the cushion can be designed independently. But care must be taken to formulate the design variables so that only manufacturable solutions are produced. Therein lies another area of work investigating the manufacture of the cushions and being able to model contact between regions of varying stiffness.

INPUT	WORK	OUTPUT
List of all factors that influence seat comfort, e.g. foam hardness, h-point, leather tension, size and shape of occupant, function, manufacture variation, etc.	Design of experiments to identify the most important factors influencing comfort.	Design variables & constraints for use in the computer optimization model and also the uncontrollable factors and their variations.

Step 5

Using the computer model generated in Step 2 together with the objective function defined in Step 3, and the design variables identified in Step 4, deterministic optimization can be carried out to generate the optimal cushion design assuming no variation in the uncontrollable factors.

INPUT	WORK	OUTPUT
Computer model from Step 2, the objective function from Step 3 and the design variables from Step 4, assuming nominal conditions, i.e. for average-sized rigid-surfaced dummy, no variation of the function e.g. driving conditions, no manufacture variation.	Carry out computerized optimization to determine the parameters that minimize the objective function.	Optimized seat for nominal conditions.

Step 6

Repeat Step 5 but apply variation of the most influential uncontrollable factors (identified in Step 4) to carry out stochastic optimization by a method such as robust design. The objective here is to determine cushions that are minimally sensitive to the variation of the uncontrollable factors thereby generating more realistic solutions.

INPUT	WORK	OUTPUT
Computer model, design variables and uncontrollable factors. Different robustness methodologies.	Robustness optimization to account for variation caused by uncontrollable factors on the objective function developed in Step 3	Robust seat that is minimally sensitive to variation in uncontrollable factors whilst being comfortable.

Step 7

Develop a more realistic deformable FE model of the human body such as the model developed by Moes *et al.* [7.25]. Using the properties of the model and using a three-dimensional scanning device the shape of a particular person can be modelled and the optimization procedure of Steps 1-7 can be repeated by replacing the dummy with the FE model of a particular person. If the design is specific to that person, as would be the case for severely deformed wheelchair user, then it will be sufficient to use a single FE model, however if the seat is being used by a variety of occupants, then variation of the occupant in the FE model will have to be accounted for. Therefore, more body shapes will need to be scanned, either directly or by making a casting of the occupant in the desired position, the latter approach has been attempted in Section 7.3.3.

The use of a more realistic FE model of the human will also mean that not only the pressure distribution but also the tissue distortion can be predicted. Therefore, the ideal objective function, described by Karg *et al.* [7.21], of minimizing the distortion of soft tissues can be used in the optimization.

7.3 PRELIMINARY WORK

The following preliminary studies have been carried out in this work:

(i) The development of an FE model of a hybrid III dummy sitting on a car seat (courtesy of Jaguar Cars Ltd.) and the corresponding prediction of the interface pressure distribution.

(ii) Pressure distribution mapping of a person sitting on a rigid surface contoured to the occupants shape.

(iii) Production of a plaster cast mould of the same person sitting down and the generation of a FE model of the casting using a three-dimensional scanning system.

7.3.1 FINITE-ELEMENT MODEL OF A DUMMY SITTING ON A CAR SEAT

In this work an FE model of a car seat and a rigid FE model of a hybrid III test dummy (courtesy of Jaguar Cars Ltd.) has been used to generate interface pressure maps between the dummy and the seat. The analysis was done using the implicit/explicit commercial FE programme LS-DYNA [7.50] which is a 'general purpose transient dynamic finite element program capable of simulating complex real world problems'. The FE mesh can be seen in Figure 7.4.

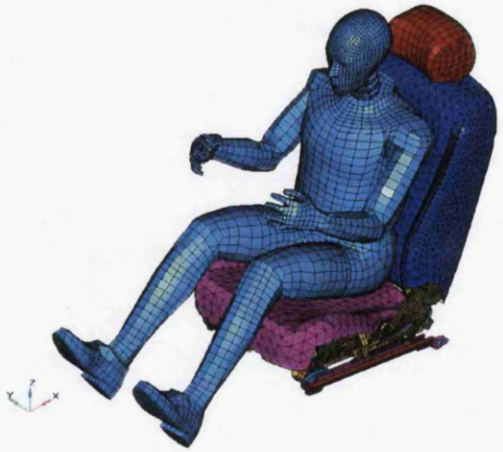


Figure 7.4: FE model of a car seat (courtesy of Jaguar Cars Ltd.) and a rigid hybrid III test dummy

Typical results of the analysis can be seen in Figure 7.5, where the pressure distribution evolves due to the dummy being lowered onto the seat. The purpose of this work was to illustrate the objective of Step 2 to produce a model of a dummy sitting on a deformable support surface and to measure the pressure distribution. The results are only indicative of the relative pressure variation and further work is required to correlate the real to the virtual pressure.

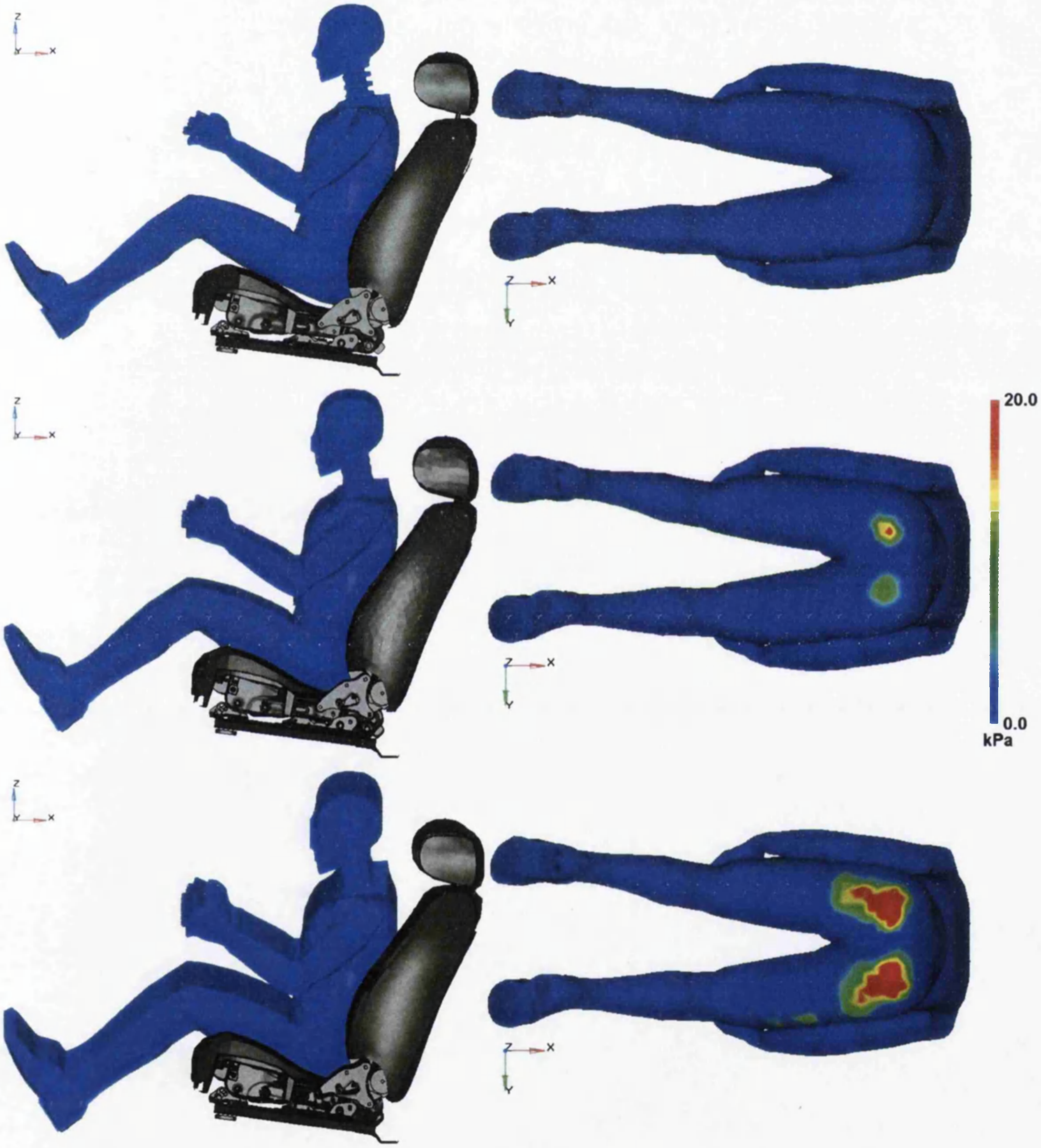


Figure 7.5: Typical FE results of the interface pressure distribution (red=high pressure, blue=low pressure) as the dummy is lowered onto the seat (top to bottom)

7.3.2 PRESSURE MAPPING

In this section a pressure mapping system developed by FSA [7.51] has been used to determine the pressure distribution of a person sitting on an almost-rigid surface contoured to the occupant's shape.

The contouring was done using a mechanism developed by the Rehabilitation unit at Morriston Hospital, Swansea. The mechanism consists of a plastic bag containing numerous, 2mm diameter polystyrene balls. The bag is placed on a support surface, in this case a hard wheelchair support system, and the occupant sits on the bag. In doing so, the balls are displaced within the bag and they mould around the shape of the occupant. Once the shape has been determined all of the air within the bag is removed using a vacuum pump. The bag containing the balls becomes almost rigid and maintains the shape of the occupant.

The next step was to place the pressure mat between the occupant and the contoured surface. The resulting pressure distributions for the wheelchair in upright and reclined positions, and with the occupant's feet on and off the footplate can be seen in Figure 7.6. The results were obtained in a crude manner but they illustrate what can be done using the pressure mapping system.

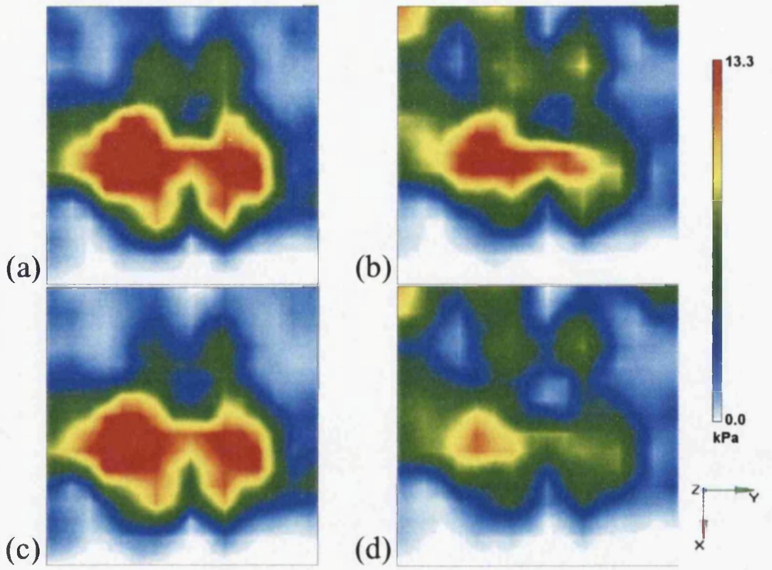


Figure 7.6: Pressure distributions (red=high pressure, blue=low pressure), obtained using the FSA pressure mapping system, for a wheelchair occupant in a variety of positions: (a) upright position with feet on the footplates; (b) upright position with feet off the footplates; (c) reclined position with feet on the footplates; (d) reclined position with feet off the footplates

7.3.3 OCCUPANT MODELLING

This stage involved making a plaster cast of the rigid mould developed in Section 7.3.2. The mould is a contoured shape of the occupant and the corresponding casting can be seen from a variety of angles in Figure 7.7.

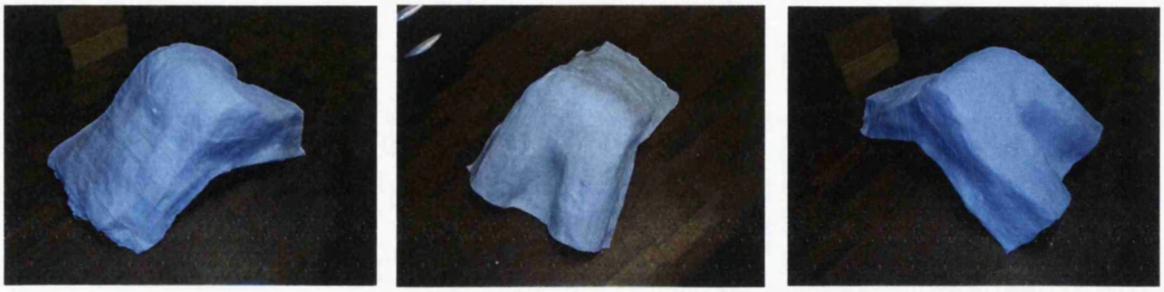


Figure 7.7: Photographs from several angles of the casting of the rigid mould developed in Section 7.3.2

The objective here was to use a three-dimensional scanning device to obtain a three-dimensional computer image of the casting. Figure 7.8 shows the use of the three-dimensional scanning device to scan the casting onto the computer. The device used here is the Modelmaker X70 attached to a Faro Goldarm, which has an accuracy of 0.084mm. The resulting data can be exported in a variety of formats including IGES and STL. The resulting FE model can be seen in Figure 7.9.

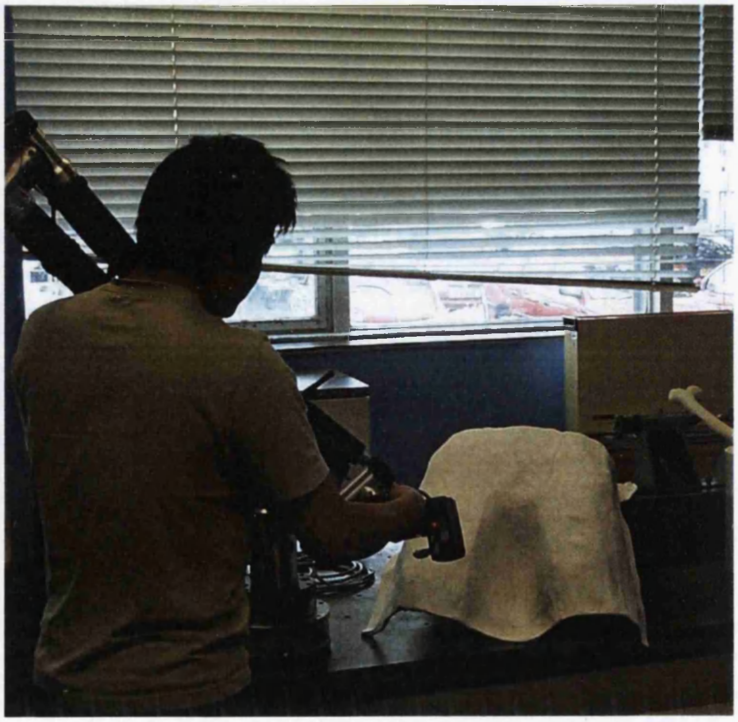


Figure 7.8: Three-dimensional scanning system being used to scan a casting of the occupant

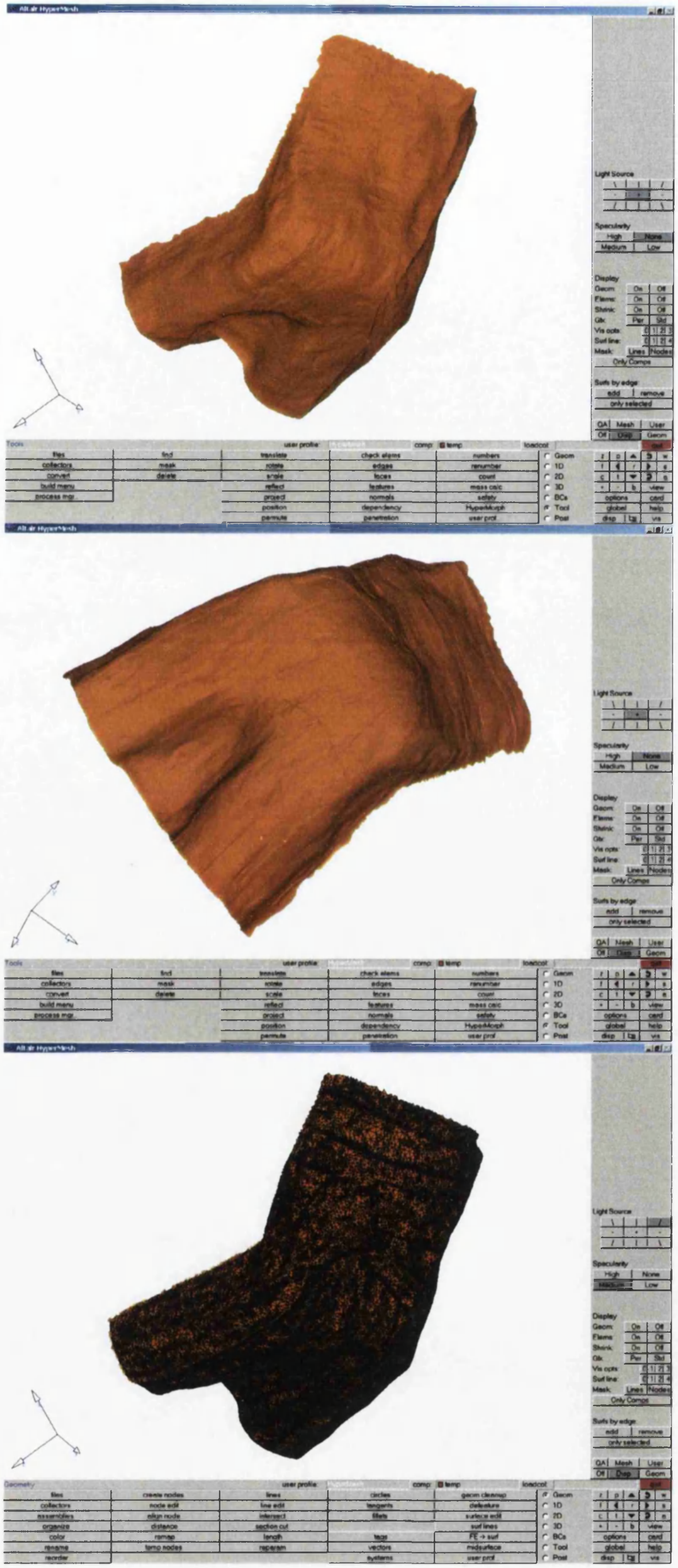


Figure 7.9: FE model of the casting from various angles with and without the FE mesh

7.4 CONCLUDING REMARKS

There is a great deal of literature regarding the design of seats for comfort. The main target of designing seats for comfort is currently in the automotive and office industry and there is a need to implement CAE into the design process. Currently very little of the technology has been implemented in wheelchair design and there is a huge market for cheap, easy to manufacture, comfortable (pressure reducing) seating, especially in developing countries, where pressure sores caused by inadequate wheelchair seating are the principal cause of death amongst disabled persons.

This chapter has reviewed the mechanisms that cause discomfort and some current CAE methods to improve comfort. A general procedure for the development of a CAE system for designing seats for comfort has been introduced. The system can be applied to the design of seats for single users and for multi-users.

7.5 REFERENCES

[7.1] Hobson D.A., "Principles of pressure management" http://www.wheelchairnet.org/WCN_WCU/SlideLectures/DAH/PM_html/PPT.html

[7.2] Schmeler M.R, Boninger M.L., Cooper R., Vitek M., "Using Peer Reviewed Literature and other Evidence to Justify Wheelchair Seating & Mobility Interventions", 18th Int. Seating Symp., 2002. http://www.seatingandmobility.ca/Iss2002/ToSunnyHill2/iss2002html/026_UsingPeerReviewedLiteratureandotherEvidence.htm

[7.3] Hobson D., Crane B., "State of the science white paper on wheelchair seating comfort", Wheelchair Seating: A State-of-the-Science Conference, 2001. http://www.rercwm.pitt.edu/RERCWM_PDF/SoSReport.pdf

[7.4] Helander M. G., Zhang L., "Field studies of comfort and discomfort in sitting", Ergonomics, 40(9), 895-915, 1997.

[7.5] Chesney D.A, Axelson P.W., Noon J.H., Siekman A.R., "RESNA '98 international cushion design competition Santa Cruz, California", Proc. of the RESNA '99 Annual Conf., Long Beach, CA, 1999. http://www.egr.msu.edu/~haddow/sig17/cushion98/Summary.pdf

[7.6] Brienza D.M., Geyer M.J., Karg P. and Jan Y.K., "The State of the Science - White Paper on Tissue integrity", Wheelchair Seating: A State-of-the-Science Conference, 2001. http://www.rercwm.pitt.edu/RERCWM_PDF/SoSReport.pdf

[7.7] Ferrarin M., Andreoni G., Pedotti A., "Comparative biomechanical evaluation of different wheelchair seat cushions", J. Rehab. Res. Dev. 37:3, May/June 2000. http://www.vard.org/jour/00/37/3/ferra373.htm

[7.8] Kosiak, M. "Etiology of Decubitus Ulcers," Archives of Physical Medicine 42:19-29, 1961.

[7.9] Ferguson-Pell M.W., "Seat cushion selection", J. Rehabil. Res. Dev, Clin Suppl2:49-73, 1990.

[7.10] Stewart S.F.C., Palmieri V., Cochran G.V.B., "Wheelchair cushion effect on skin temperature, heat flux, and relative humidity." Arch. Phys. Med. Rehabil., 61:229-33, 1980.

[7.11] Finestone H.M., Levine S.P., Carlson G.A., Chizinsky K.A., Kett R.L., "Erythema and skin temperature following continuous sitting in spinal cord injured individuals." J. Rehabil. Res. Dev., 28(4):27-32, 1991.

[7.12] Bennet L., Kavner D., Lee B.K., Trainor F.A., "Shear vs. pressure as causative factors in skin blood flow occlusion.", Arch. Phys. Med. Rehabil., 60:309-14., 1979.

[7.13] Springle S., Chung K., Brubaker C.E., "Reduction of sitting pressures with custom contoured cushions.", J. Rehabil. Res. Dev., 27(2):135-40, 1990.

[7.14] Garber S.L., Krouskop T.A., "Wheelchair cushion modification and its effect on pressure." Arch. Phys. Med. Rehabil., 65:579-83, 1984

[7.15] Patterson R.P., Fisher S.V., "Sitting pressure-time patterns in patients with quadriplegia." Arch. Phys. Med. Rehabil., 67:812-4., 1986.

[7.16] Levine S.P., Kett R.L., Ferguson-Pell M. "Tissue Shape and Deformation Versus Pressure as a Characterization of the Seating Interface," Assistive Technology 2: 93-9, 1990.

[7.17] Brienza D.M., Karg P.E, Geyer M.J., Kelsey S., Trefler E. "Relationship between pressure ulcer incidents and buttock seat cushion interface pressure in at-risk elderly wheelchair users" Archives of Physical Medicine & Rehabilitation, 82, pp.529-533, 2001.

[7.18] Geyer M.J., Brienza D.M., Karg P., Kelsey S., Trefler E., "Are commercial seat cushions efficacious in preventing pressure ulcers?" Proc. of RESNA Conf., Orlando, FL, June 28-July 2. pp 369-371, 2000.
http://www.wheelchairnet.org/WCN_WCU/SlideLectures/MJG/GeyerCushion.pdf

[7.19] Conine T.A., Herschler C., Deachsel D., Peel C., Pearson A. "Pressure ulcer prophylaxis in elderly patients using polyurethane foam or Jay wheelchair cushions", Int. J. of Rehabilitation Research, 17. pp.123-137, 1994.

[7.20] Shapcott N., Levy B., "By the numbers", TeamRehab Report, 10(1), pp.16-21, Jan 1999.http://www.wheelchairnet.org/WCN_ProdServ/Docs/TeamRehab/RR_99/Jan_99/9901art1.PDF

[7.21] Karg T.E., Brienza D.M., Brubaker C.E., Wang J., Lin C.T. "A system for the design and analysis of seat support surfaces", Proc. of the RESNA '96 Annual Conf., Salt Lake City, Utah, June 7-12, 1996.
http://www.dinf.ne.jp/doc/english/Us_Eu/conf/resna96/page90.htm

[7.22] Personal communication with Jaguar Cars Ltd.

[7.23] Brienza D.M., Chung K.C., Brubaker C.E., Wang J., Karg T.E., Lin C.T., "A system for the analysis of seat support surfaces using surface shape control and simultaneous measurement of applied pressure", IEEE Trans Rehabil Eng 1996; 4(2):103-13. http://www.dinf.ne.jp/doc/english/Us_Eu/conf/resna96/page91.htm

[7.24] Wang J., Brienza D.M., Yuan Y., Karg P., Xue Q., "A compound sensor for biomechanical analyses of buttock soft tissue in vivo", J. Rehabil. Res. Dev., Jul-Aug 37:433-43, 2000. http://www.vard.org/jour/00/37/4/wang374.htm

[7.25] Moes C.C.M., Horváth I. "Finite Elements Model of the Human Body: Geometry and Non-linear Material Properties", Proceedings of the TMCE 2002, April 22-26, Wuhan, China, 2002. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/tmce2002moes.pdf

[7.26] Moes C.C.M., and Horváth, I., "Ergonomics considerations for the conceptualization of the shape of body supports", Proceedings of the 1999 International Design Seminar on Integration of Process Knowledge into Design Support Systems, pages 435–448, Enschede, 1999. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/cirp_99_Moes.pdf

[7.27] Moes C.C.M., "Pressure Distribution and Ergonomics Shape Conceptualization. In D. Marjanovic (Ed.), Proceedings of the 6th International Design Conference, Zagreb, pp. 233–240, 2000.
http://dutoce.io.tudelft.nl/~jouke/docdb/docs/design2000_moes_pressure.pdf

[7.28] Moes C.C.M., "Geometric Model of the Human Body." In I. Horváth, A. J. Medland, & J. S. Vergeest (Eds.), Proceedings of the TMCE2000 Delft, the Netherlands: Third International Symposium on Tools and Methods of Competitive Engineering Delft University Press pp. 79–92, 2000.
http://dutoce.io.tudelft.nl/~jouke/docdb/docs/tmce2000_moes.pdf

[7.29] Moes C.C.M., Rusák Z., Horváth I. "Application of vague geometric representation for shape instance generation of the human body" Proceedings of DETC'01, Computers and Information in Engineering Conf., Pittsburgh, ASME 2001, CDROM:DETC2001/CIE–21298, 2001. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/asme2001moes.pdf

[7.30] Moes C.C.M., "Generation of Shape Instances for FE Modelling of the Human Body" In Proceedings of the Int. Conf. on Engineering Design ICED2001, Glasgow, Aug 21–23, 2001. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/iced2001moes02.pdf

[7.31] Seitz, T. and Bubb, H., Measuring of Human Anthropometry, Posture and Motion, In: Proceedings of SAE Conference on Human Modelling, The Hague, 18-20 May, Technical Paper Series No. 1999-01-1913, SAE International, 1999. http://www.ergonomie.tum.de/~seitz/SAE1999_HumanModelling.pdf

[7.32] Moes C.C.M., Horváth I. "Using Finite Elements Model of The Human Body for Shape Optimization of Seats: Optimization Material Properties", Design 2002. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/design2002moes.pdf

[7.33] Moes C.C.M., Horváth I. "Optimizing the Product Shape for Ergonomics Goodness Index. Part I: Conceptual Solution", ES 2002. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/ergonomics2002moesa.pdf

[7.34] Moes C.C.M., Horváth I. "Optimizing the Product Shape for Ergonomics Goodness Index. Part II: Elaboration for Material Properties", ES 2002. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/ergonomics2002moesb.pdf

[7.35] Manschot J.F.M., Brakkee A.J.M., "Characterization of in vivo mechanical skin properties independent of measuring configuration", Bioengineering and the Skin, 3:1–10, 1987.

[7.36] Moes C.C.M., "Mathematics and Algorithms for Pressure Distribution Controled Shape Design", Methods for Performance and Sustainability, Proc. of the 13th Int. Conf. on Engineering Design, ICED01 pp 99-106, Glasgow, UK, 2001. http://dutoce.io.tudelft.nl/~jouke/docdb/docs/iced2001moes01.pdf

[7.37] Rasmussen J., Dahlquist J., Damsgaard M., de Zee M., Christensen S. T., "Musculoskeletal modeling as an ergonomic design method." Int. Ergonomics Assc. XVth Triennial Conf., Seoul, August 24-29, 2003. http://anybody.auc.dk/pdf/Seoul.pdf

[7.38] Mandal A.C., "The seated man (homo sedens) - The seated work position. Theory and practice", Applied Ergonomics, 12(1), pp19–26, 1981.

[7.39] Zhang, L., Helander, M.G., Drury, C.G., "Identifying Factors of Comfort and Discomfort in Sitting". Human Factors, 38(3), pp377-389, 1996. http://dutoce.io.tudelft.nl/~jouke/docdb/

[7.40] Kolich M., "Automobile seat comfort: occupant preferences vs. anthropometric accommodation", Applied Ergonomics, 34(2), pp.177-184, March 2003.

[7.41] Seidl, A., "Man model RAMSIS - Analysis, synthesis and simulation of three-dimensional postures of humans", PhD thesis, Munich, 1993.

[7.42] Lengsfeld M., Frank A., van Deursen D.L., Griss P., "Lumbar spine curvature during office chair sitting", Medical Engineering & Physics 22(9) pp. 665-669, 2000.

[7.43] Stelze W., "Simulating comfort before the car exists", Auto Technology 4, 2002.

[7.44] Schmale G., Stelzle W., Kreienfeld Th., Wolf C.D., Härtel Th., Jödicke R., "Cosyman - A Simulation Tool for Optimization of Seating Comfort in Cars" SAE digital human modelling conf., Munich, pp. 301-311, ISBN 0-7680-1025-X, June 2002.

[7.45] http://www.johnsoncontrols.com/comfortlab/frameset5.htm

[7.46] Flight International magazine, "Air-cushion comfort", pp.28, Flight International magazine, 27th May- 2ndJune, 2003.

[7.47] Pankoke S., Balzulat J., Wölfel H. P., "Vibrational comfort with CASIMIR and RAMSIS using a dynamic finite-element model of the human body", SAE digital human modelling conf., Munich, pp. 146-157, ISBN 0-7680-1025-X, June 2002.

[7.48] Bartels V.T., "Thermal comfort of aeroplane seats: influence of different seat materials and the use of laboratory test methods", Applied Ergonomics, Volume 34, Issue 4, pp. 393-399, July 2003.

[7.49] Huizenga C., Hui Z., Arens E., "A model of human physiology and comfort for assessing complex thermal environments", Building and Environment, Volume 36, Issue 6, pp.691-699, July 2001.

[7.50] "LS-DYNA: Non-linear Dynamic Analysis of Structures in Three dimensions", LSTC, Version 960, 2002.

[7.51] Vista Medical Ltd, FSA pressure mapping system.

CHAPTER 8

SEA TRANSPORTATION OF CARGO: A CRASH SIMULATION USING FINITE ELEMENT ANALYSIS

Summary: This chapter is concerned with developing a finite element (FE) computer simulation of a cargo inside its transportation packaging. The cargo is a complex multi-body system. The assembly relies on friction forces due to gravitation and belt tension to secure the components. The cargo is subjected to a time-dependent, multi-axial loading. The research involved modelling the contact between the components; the formulation of the load curves that model the motion of the sea; simulation of the loadings related to the assembly and securing of the transformer packaging; the formulation of realistic loading regimes to simulate the sequence in which all of the loadings are applied in reality; identifying the interactions and the load path sequence for several different loading regimes. Using the model, potential problem areas in the packaging used to hold the cargo in place during sea transportation have been identified by applying various load cases, simulating assembly, rotation and/or acceleration in pre-defined directions. Conservative friction factors were used so that, if failure of the packaging design occurs with these values, no further computations with lower values need to be carried out. Concentrating on two load cases, the computations using these assumptions show that the transformer moves under normal conditions when it is accelerated along the x-axis with an acceleration of 0.62g, and when the transformer is subjected to a rotation of 12.6°. Both cases can happen on a typical cargo ship. The movement causes cyclic loading on the bolts (+128MPa/0MPa at 0.62g) connecting the top yoke with the flitch plates. It can be concluded that the design of the transformer packaging is unsuitable for sea transport.

8.1 INTRODUCTION

This chapter is set out as follows:

Section 8.1:	Gives an introduction to the problem, identifying the key areas of research that have been carried out.
Section 8.2:	Gives an overview of the FE model.
Section 8.3:	Looks at the inter-component contact modelling.
Section 8.4:	Identifies the loading conditions due to, (i) the motion of the sea and (ii) the packaging of the transformer. Appropriate loading regimes are then determined.
Section 8.5:	Describes the loading time histories used in seven crash models to predict the behaviour of the transformer and its packaging, under a variety of loading regimes.
Section 8.6:	Identifies the inter-component interactions and the load path sequence for several of the crash models described in Section 8.5.
Section 8.7:	Gives some concluding remarks.
Section 8.8:	Lists the references used in the chapter.

In previous chapters the focus of the work has been mainly concerned with optimization modelling and simulation. This corresponds to the second and third columns of the three-column concept in Figure 1.3. This chapter is focused solely on the third column of the three-column concept with the development of a robust simulation to predict the behaviour of a complex multi-body system.

The work involves the development of a FE computer simulation of a cargo inside its transportation packaging. For this problem, carrying out physical simulations is infeasible since the cargo is extremely expensive. Furthermore, identifying the interactions and the load path sequence for several different loading regimes would be very difficult, unlike with a computer simulation, where it is relatively simple.

Using the simulation, a crash analysis is used to investigate the behaviour of a 139 tonne electrical transformer and its transportation packaging, during sea transportation. The transformer and its packaging, consists of 46 main components that interact in a highly complex manner, meaning that, intuitive prediction of the behaviour of the entire model is non-trivial. Contact forces, inertia forces, sea-motion-induced forces and packaging-induced forces influence the interactions. In order to accurately predict the local and global behaviour of the transformer in its transportation state, they all need to be included in the model.

For the purpose of this study, a non-linear FE model has been developed to simulate the effect of the sea's motion on the transformer and its packaging. The analysis is done using the implicit/explicit commercial FE programme LS-DYNA [8.1]. In addition to developing the model from the engineering drawings, the research has involved the following key areas:

(i) Modelling the contact between the components.

(ii) Formulation of the load curves that model the motion of the sea.

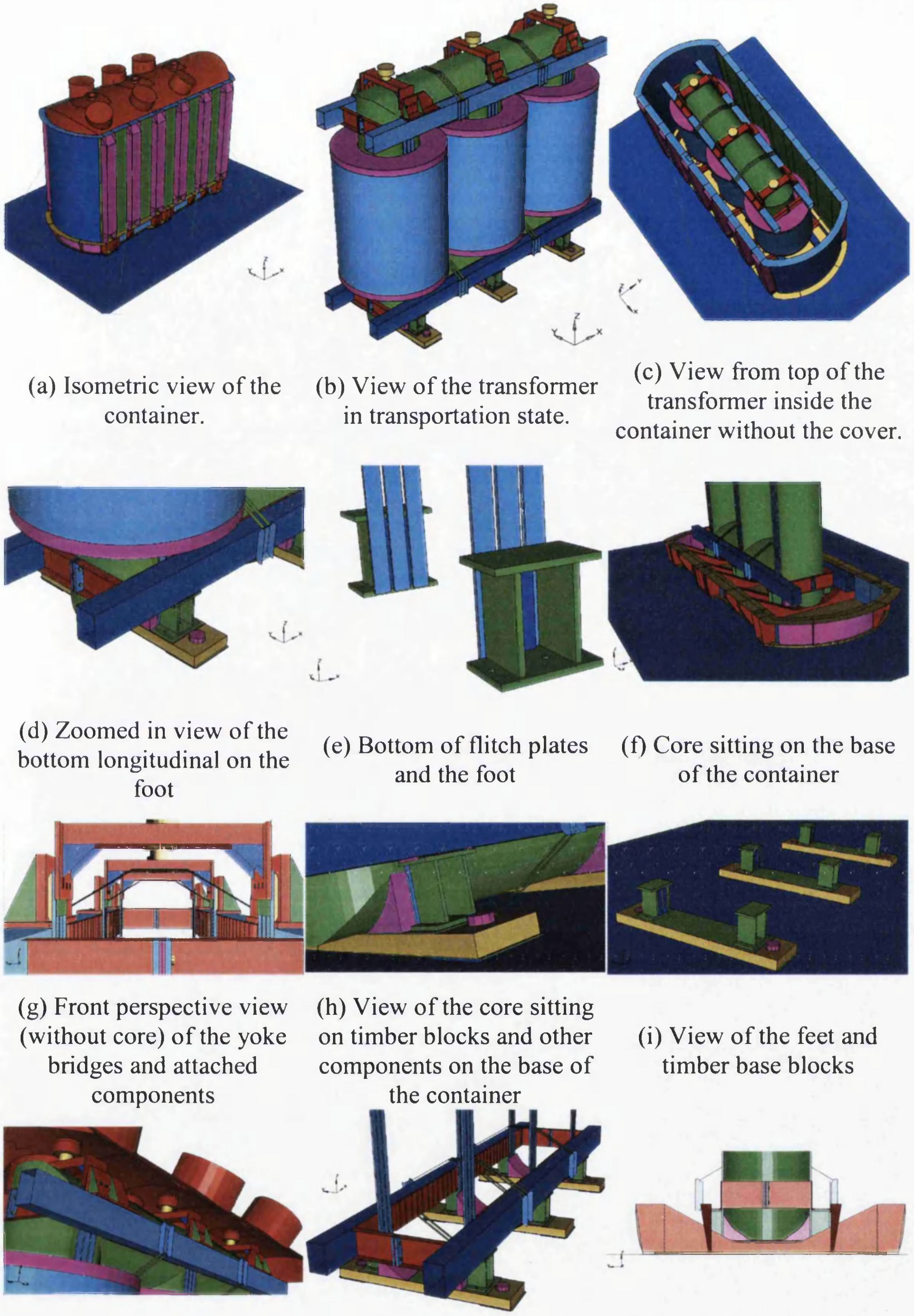
(iii) Simulation of the loadings related to the assembly and securing of the transformer packaging. This involves modelling the pre-tensioning and compression of the bolts.

(iv) The formulation of realistic loading regimes to simulate the sequence in which all of the loadings are applied in reality.

(v) Identifying the interactions and the load path sequence for several different loading regimes.

8.2 GENERAL OVERVIEW OF THE FINITE ELEMENT MODEL

To carry out the "crash-analysis", a full FE-model was constructed. An overview of the FE model is presented here. Figures 8.1 and 8.2 give an illustrative overview of the model. Figure 8.1 illustrates the model from various angles (beneath each image, a short description is given). Figure 8.2 shows the whole model 'clipped' in the z-plane at various positions. The final model consists of 46 Components and 288001 elements including: 99970 hex8 solid elements, 95286 quad4 shell elements, 86053 contact elements, 3096 Penta6 solid elements, 2078 welds, 1192 tria3 shell elements, 198 rigid elements, 120 bar2 elements and 8 rod elements.



(a) Isometric view of the container.

(b) View of the transformer in transportation state.

(c) View from top of the transformer inside the container without the cover.

(d) Zoomed in view of the bottom longitudinal on the foot

(e) Bottom of flitch plates and the foot

(f) Core sitting on the base of the container

(g) Front perspective view (without core) of the yoke bridges and attached components

(h) View of the core sitting on timber blocks and other components on the base of the container

(i) View of the feet and timber base blocks

(j) View of the yoke bridges slotting into cover

(k) View showing bottom part of the transformer packaging without core.

(l) front view of container base showing base gussets with the core sitting on the timber blocks.

Figure 8.1: Images of the FE model from various angles

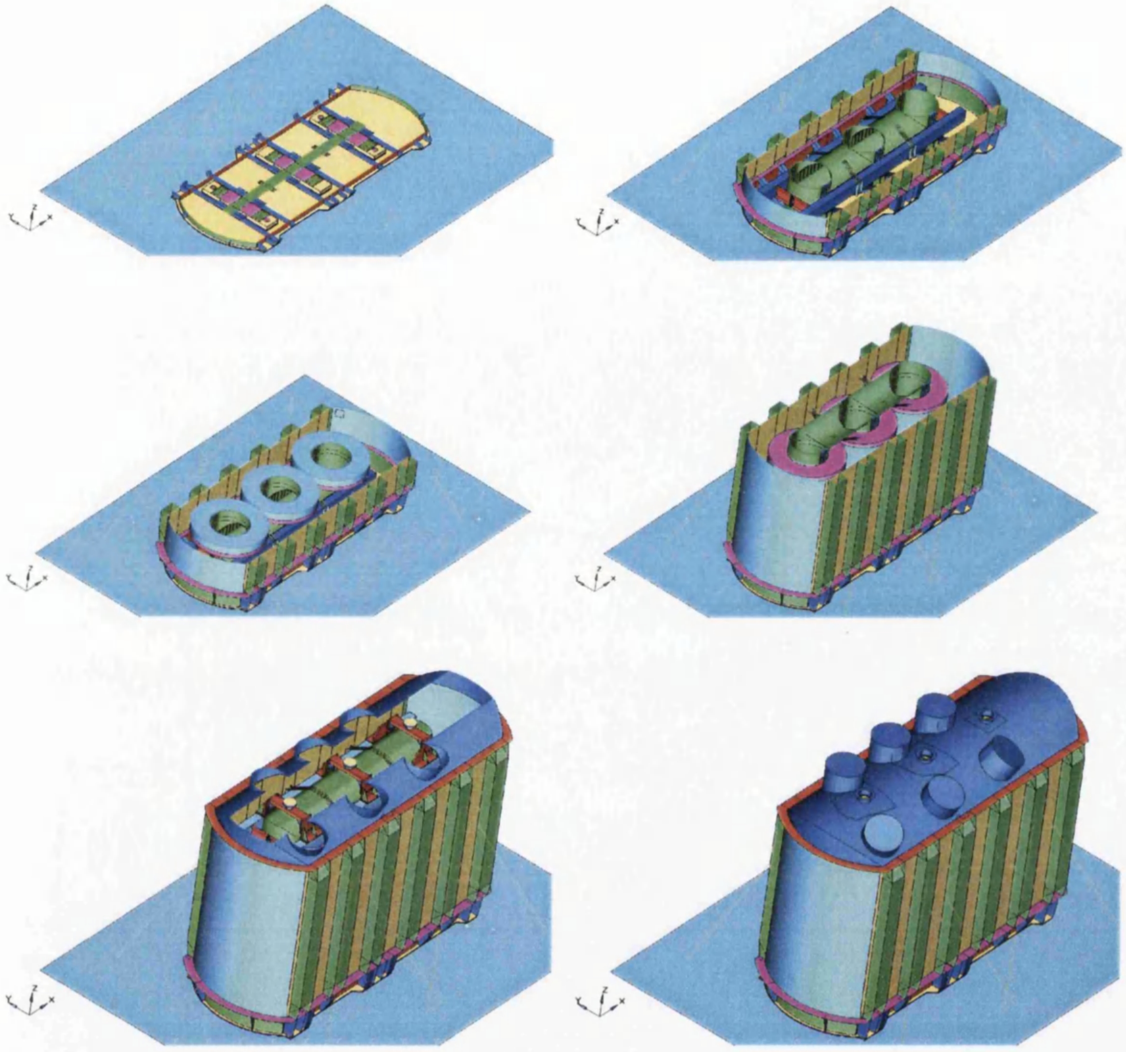


Figure 8.2: The entire model 'clipped' in the z-plane at various positions

8.3 MODELLING CONTACT BETWEEN COMPONENTS

In order to model the interface contact, the static and the dynamic coefficients of friction are required. Their determination is difficult, especially as the experimental values are influenced not only by the surfaces in contact, but also by other factors such as "moisture, traces of grease, dust, deformation of paint layer and vibration" [8.2]. There are two principal sources of vibration on a ship: the engine and the propeller. Usually the engine is isolated to prevent transmission of vibrations to the ship. However, the vibrations from the propeller (blade by-pass frequency is approximately 10Hz) are transmitted to the ship's hull and there to any cargo in the ship's loading bays. When a transformer is tested, it is usually submerged in oil which is drained during transportation, and so, it is highly likely that traces of oil will remain on the surfaces. The vibration may cause the oil mist to get between the contact surfaces, causing the coefficients of friction to reduce, especially between the timber and steel surfaces [8.3].

In this model there are three interface types and, based upon Refs. [8.2, 8.3, 8.4], conservative values of the static, μ_S and dynamic, μ_D friction coefficients are chosen as:

- painted steel to painted steel contact: μ_S=0.28, μ_D=0.19
- bare steel to timber contact: μ_S=0.15, μ_D=0.15
- painted steel to timber contact: μ_S=0.28, μ_D=0.19

By choosing conservative values, the behaviour predicted by the model will be exacerbated in reality if the friction coefficients are lower. The principal contact areas are shown in Figure 8.3.

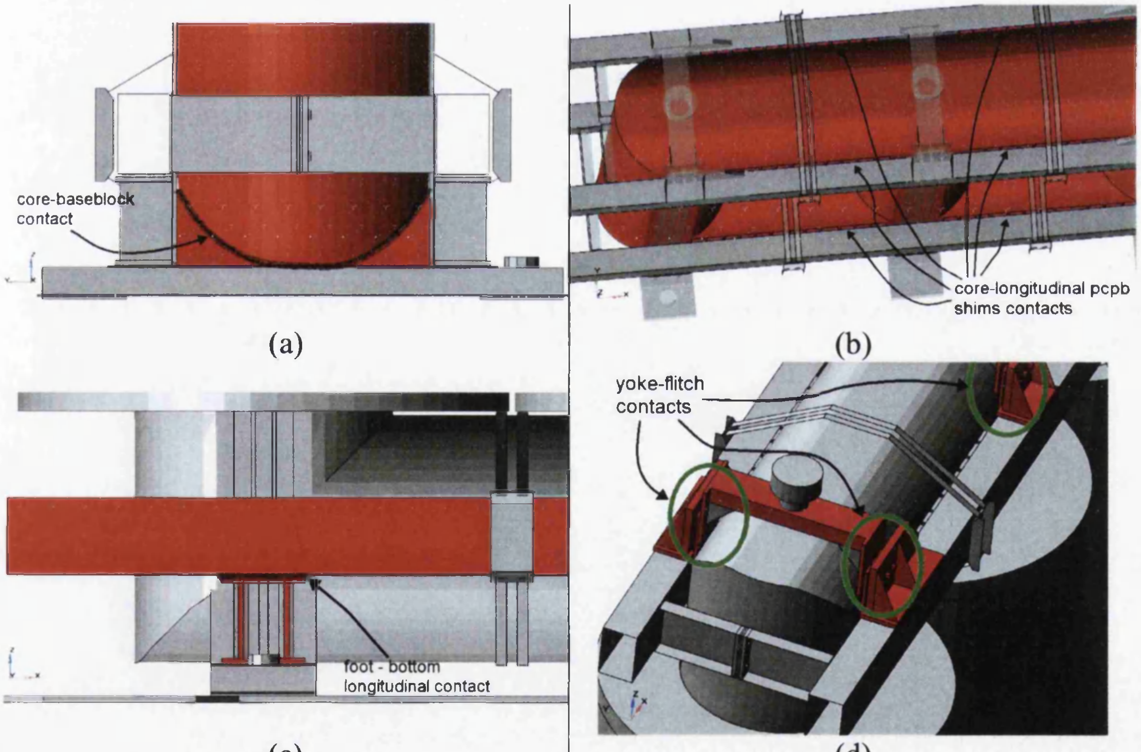

core-baseblock contact
core-longitudinal pcpb shims contacts
(a)
(b)
yoke-flitch contacts
foot - bottom longitudinal contact
(c)
(d)

Figure 8.3: Principal contact areas: (a) The bare steel-to-timber contact between the core and the baseblock. (b) The bare steel-to-timber contact between the longitudinal timber shims and the core. (c) The painted steel to timber contact between the bottom longitudinal, the locating bars and the timber layer on the foot. (d) The contact between the yoke bridge, flitch plates, yoke_flitch bolts and the timber shims

8.4 LOADING CONDITIONS

In addition to the sea-induced loading conditions, there are also additional loads related to the assembly of the transformer packaging. These are known as clamping and sizing forces and are applied to secure everything together. In order to apply loadings, load curves defining the time history of the loading are defined. This section gives details of the load curves.

8.4.1 GRAVITY LOADING

Gravity is applied to the model applying a body acceleration of $9810\text{mm}/\text{s}^2$ to each element in the positive z-direction.

8.4.2 PACKAGING-INDUCED LOADINGS

8.4.2.1 SIZING FORCE

The sizing force is applied as nodal-point forces to the windings and the top longitudinal as can be seen in Figure 8.4, where the windings and bottom longitudinal are separated using rigid bodies.

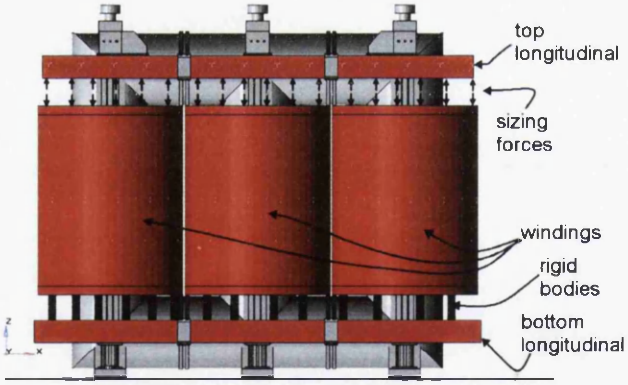


Figure 8.4: Sizing force

The value of each nodal force is calculated as follows:

- The total sizing force per winding is 816kN
- On each winding there are 12 nodal forces
- Therefore each nodal force is 816/12 = 68kN

In total there are 72 nodal 'sizing' forces of magnitude 68kN. There are 36 forces acting normal to the three windings and 36 acting normal to the top longitudinal as can be seen in Figure 8.4.

8.4.2.2 Clamping forces

There are three clamping forces used in the model:

1) Endstop clamping force (Figure 8.5)
2) Longitudinals to core clamp force (Figure 8.6)
3) Yoke to Flitch clamp force (Figure 8.7)

In order for the direction of the forces to be correctly orientated the forces are applied as pressure forces.

8.4.2.2.1 Endstop clamping force

There are 8 endstop bolts in total, each with a clamping force of 145kN applied. The location of the endstop clamping force is shown in Figure 8.5(a). To model this force an equivalent pressure is applied to the faces of the elements surrounding the bolt hole as can be seen by the white area in Figure 8.5.

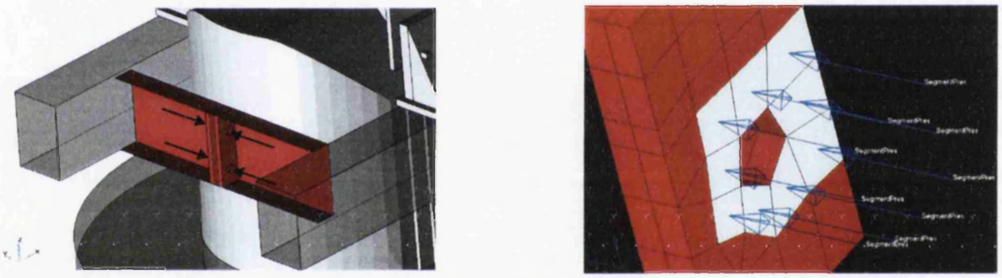


Figure 8.5: End stop clamping force: (a) location in the model; (b) force modelling

The pressure is applied to the centroid of each of the element faces and is calculated as follows:

$$\text{force} = 145\text{kN}$$

$$\text{area (white region)} = 3094.496\text{mm}^2$$

$$\text{pressure} = \frac{145\times10^3}{3094.496} = 46.875\text{MPa}$$

8.4.2.2.2 Longitudinals to core clamp force

The tensioning of the yoke straps has the effect of clamping the longitudinals to the core as shown in Figure 8.6(a). The specified tensioning per strap is 101.5kN. This is modelled by applying pressure forces to the white areas of yoke strap bracket in Figure 8.6(b) below.

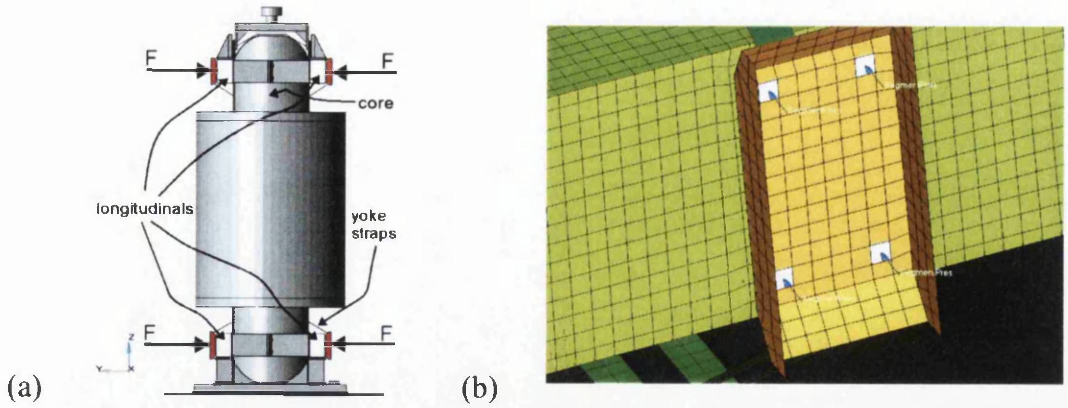


Figure 8.6: Longitudinals to core clamp force: (a) location in the model; (b) force modelling

There are 4 straps per bracket. So the equivalent force F in Figure in 8.6(a)), is $4 \times 101.5\text{kN} = 406\,\text{kN}$. The pressure is applied to the centroid of each of the element faces and is calculated as follows:

$$\text{force} = 406\text{kN}$$

$$\text{area (white regions)} = 1616.592\text{mm}^2$$

$$\text{pressure} = \frac{406 \times 10^3}{1616.592} = 251.15\text{MPa}$$

8.4.2.2.3 YOKE_FLITCH CLAMP FORCE

The yoke_flitch is a critical area of the transformer packaging and both the clamping force caused by the bolt and pre-tensioning of the bolts is required. Figure 8.7 shows the location of the yoke_flitch bolts. There are 18 yoke_flitch bolts in total, each with a pre-tensioning force of 145kN and a clamping force of 145kN force.

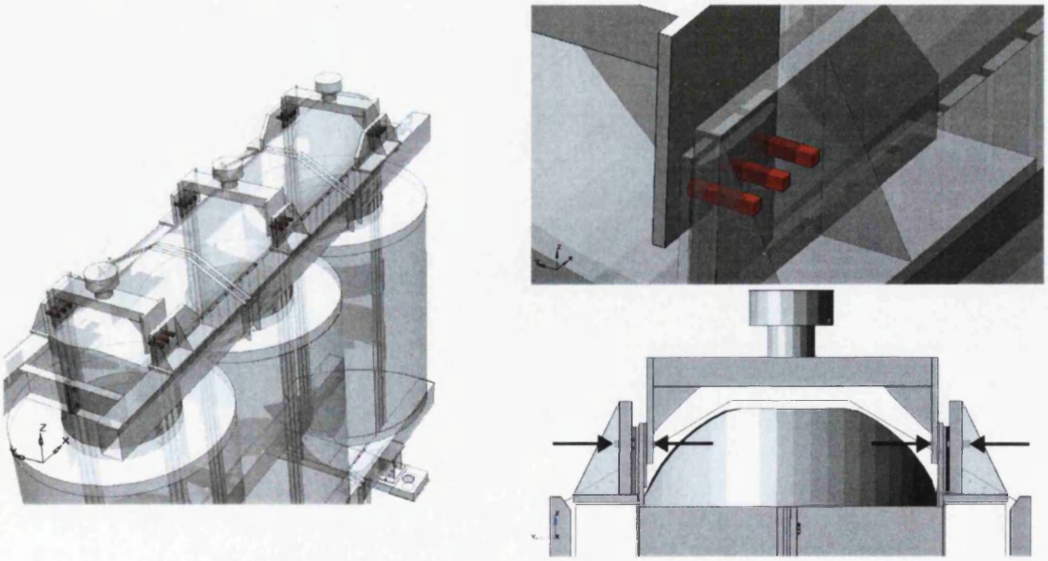


Figure 8.7: yoke_flitch bolts location and the direction of the clamping

To model this an equivalent pressure is applied to the faces of the elements surrounding the bolt hole, as can be seen by the white area in Figure 8.8. The pressure is applied to the centroid of each of the element faces, and is calculated as follows:

force = 145kN

area (white region) = 3231.277mm^2

$$\text{pressure} = \frac{145 \times 10^3}{3231.277} = 44.874\text{MPa}$$

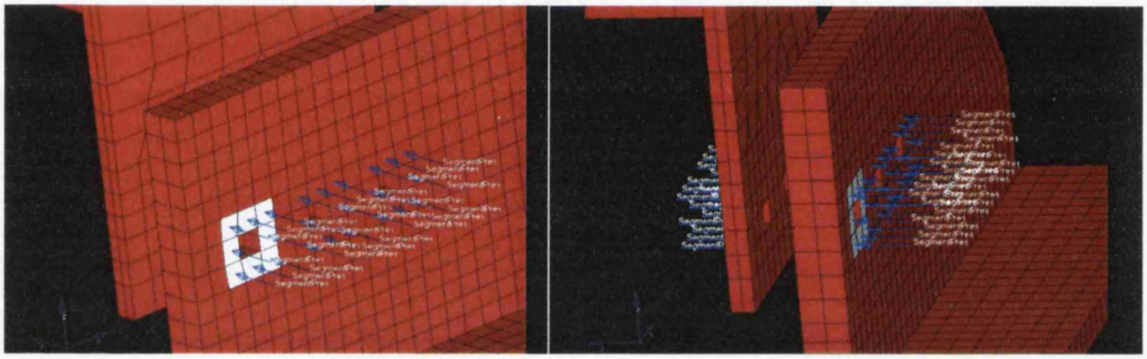


Figure 8.8: yoke_flitch clamp force

Note that the areas for each bolt are slightly different from the white area shown in Figure 8.8, this has been accounted for in the model. To model the pre-tensioning an equivalent pressure force is applied to the faces of the yoke_flitch bolts as can be seen by the white area in Figure 8.9. The pressure is applied to the centroid of each of the element faces and is calculated as follows:

force = 145kN

area (white region) = 400.0mm^2

$$\text{pressure} = \frac{145 \times 10^3}{400.0} = 362.5\text{MPa}$$

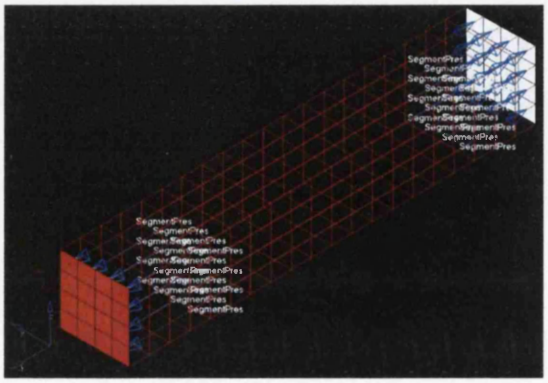


Figure 8.9: yoke_flitch bolt tension force

8.4.3 SEA-INDUCED LOADINGS

The motion of the sea causes accelerations and rotations of the ship. The ship's deck, shown in Figure 8.10, is modelled using rigid elements and the base of the container tank is attached to it at several points using rigid body elements. The rotations and accelerations are applied to the FE model by rotating and accelerating the deck.

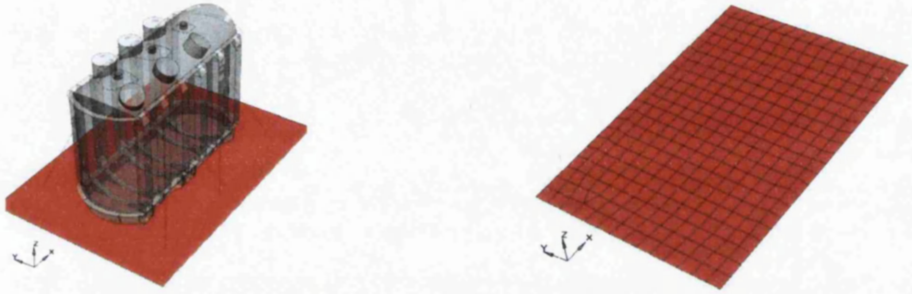


Figure 8.10: FE model of the deck of the ship

In order to model the rotations induced by the motion of the sea, the maximum pitch and roll angles of the vessel are required. From Ref. [8.5] two semi-empirical equations for identifying the maximum pitch and roll angles are:

$$\sin\phi_{rm} = \left(0.45 + 0.1\left(\frac{L}{B}\right)\right)\times\left(0.54 - \left(\frac{L}{1270}\right)\right) \tag{8.1}$$

$$\phi_{pm} = 12\,\mathrm{e}^{-0.0033L} \leq 8^{\circ} \tag{8.2}$$

where ϕ_{rm} is the maximum roll angle, ϕ_{pm} is the maximum pitch angle, L is the ship length and B is the width of the ship. It is assumed that the length of the ship is 100m and the width 20m, meaning that, ϕ_{rm} is 26° and ϕ_{pm} is 8°. A schematic diagram of the pitch and roll angles, and the orientation of the transformer in the ship, is shown in Figure 8.11.

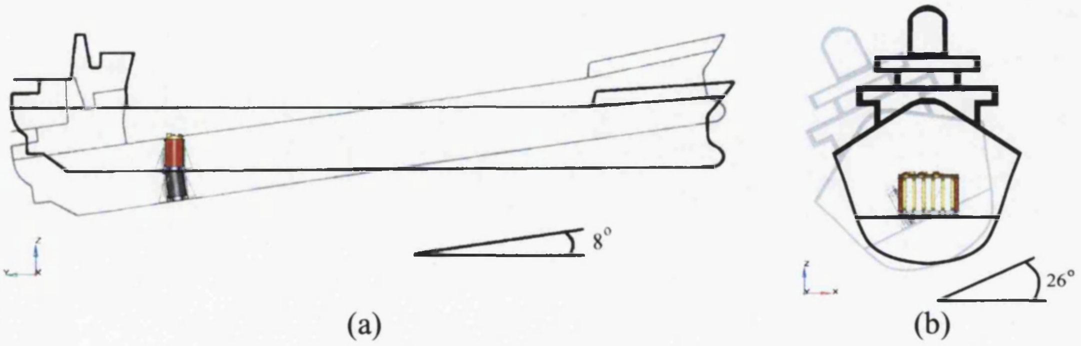


Figure 8.11: Schematic diagram showing the transformer orientation and the ship rotating from 0° (dark outline) to the maximum angles (lighter outline) for (a) pitching and (b) rolling

The motion of the sea is assumed to be sinusoidal, with a wave period of 12 seconds [8.5]. Then the function used to model the sea's motion, $f(t)$, is as follows:

$$f(t) = \phi_{\max} \sin\left(\frac{2\pi t}{T}\right) \tag{8.3}$$

where $\phi_{\max}$ is the maximum angle in radians, t is the time and T is the wave period. The corresponding rotation curves are shown in Figure 8.12 where the rotation about the y-axis corresponds to the roll and the rotation about the x-axis corresponds to the pitch. In order to apply this motion to the FE model, it is sufficient to apply the rotation for the first three seconds of rotation and the remainder of the motion is not computed, this is the grey region in Figure 8.11.

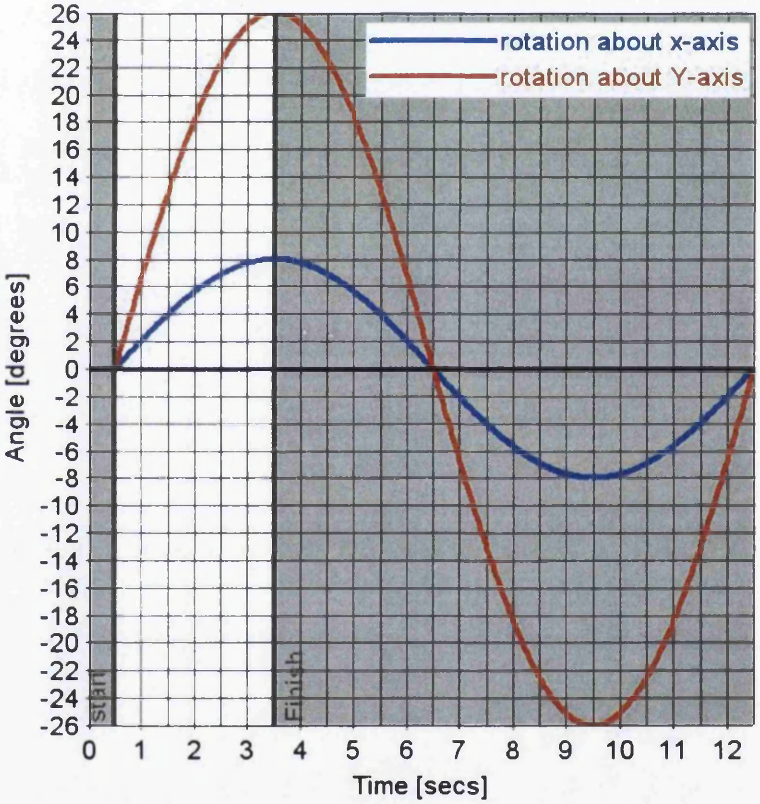


Figure 8.12: Rotation curves for rotation about x-axis (blue) and y-axis (red) for 12 seconds with grey regions not computed

A realistic maximum value of acceleration, induced by sea motion, is 3g [8.5]. In order to model the loadings imparted by this acceleration, a linear acceleration from rest to the assumed maximum of 3g is imposed on the deck using the equivalent velocity curve. It is assumed that the acceleration is over a period of 1.25 seconds. Figure 8.13(a) shows the acceleration curve and (b), the corresponding velocity curve.

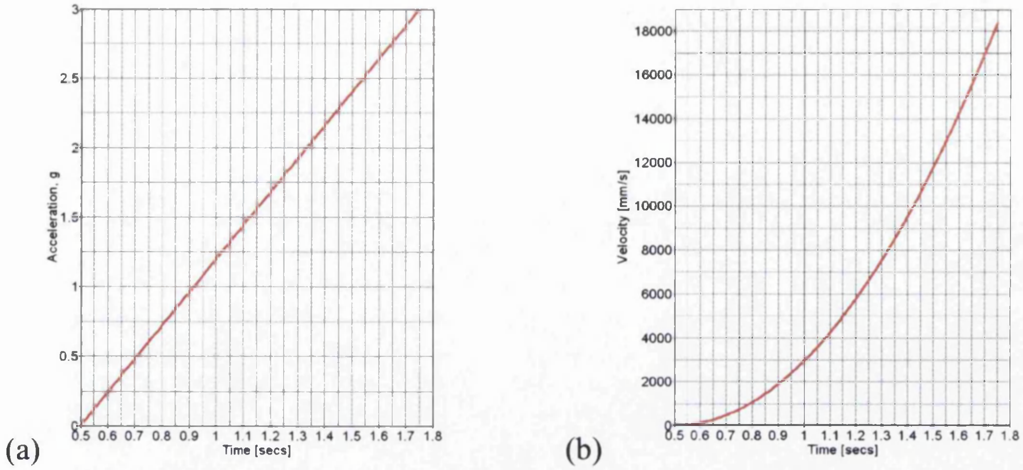


Figure 8.13: Load curves for (a) the assumed acceleration and (b) the corresponding velocity curve

For each model a time history of the loading regime is given. Particular attention is paid not only to the load history during the transport, but also the sequence in which the gravity loading and the packaging-induced clamping and sizing forces are applied. The sequence is shown in Figure 8.14. The implication of this is that, in addition to simulating the acceleration and rotation during transportation, the application of the clamping forces, the sizing forces and the gravity loading were also simulated for the stationary transformer as the packaging is being assembled. Only after this "assembly simulation", the real "crash analysis" with the given load history is started.

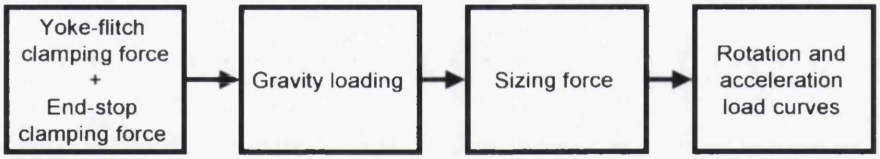

Yoke-flitch clamping force + End-stop clamping force
Gravity loading
Sizing force
Rotation and acceleration load curves

Figure 8.14: Model loading sequence

This procedure ensured a realistic model behaviour, but added considerably to the computing time necessary to complete one analysis (typically 10 CPU days on a Compaq ES40 server with 4 processors for the non-rotated models). For the cases with rotation of the model further computations needed to be carried out, adding more computing time to the solution of one case.

The model assumes ideal assembly conditions for the transformer and neglects effects of "real" assembly leading to uneven pressure distributions. The "real" transformer is likely to be less resistant to outside loading than the ideal model.

8.5 MODEL DESCRIPTIONS

In total, seven 'crash-analysis' models with various loadings, were run:

Model number	
1	x acceleration
2	y acceleration
3	+z acceleration
4	combined x and y rotation
5	x acceleration with loose bolts
6	y rotation with loose bolts
7	y rotation

Table 8.1: Model descriptions

The yoke_flitch bolts, shown in Figure 8.7, may loosen under cyclic loading due to the motion shown in Figure 8.15 caused by positive and negative accelerations in the x-direction. The loosening of the yoke_flitch bolts will lead to loss of support for the core and the top frame by the top yoke. To model this the yoke_flitch clamp force and the sizing force are removed in Models 5 and 6.

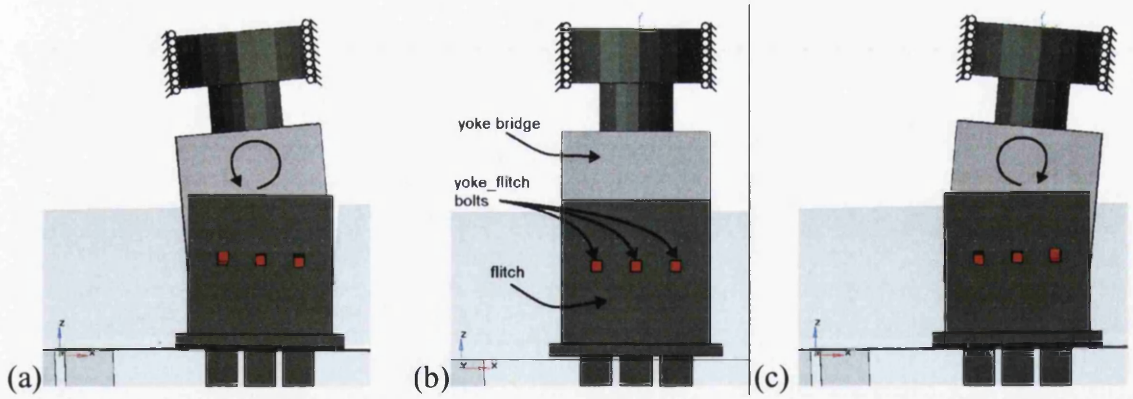
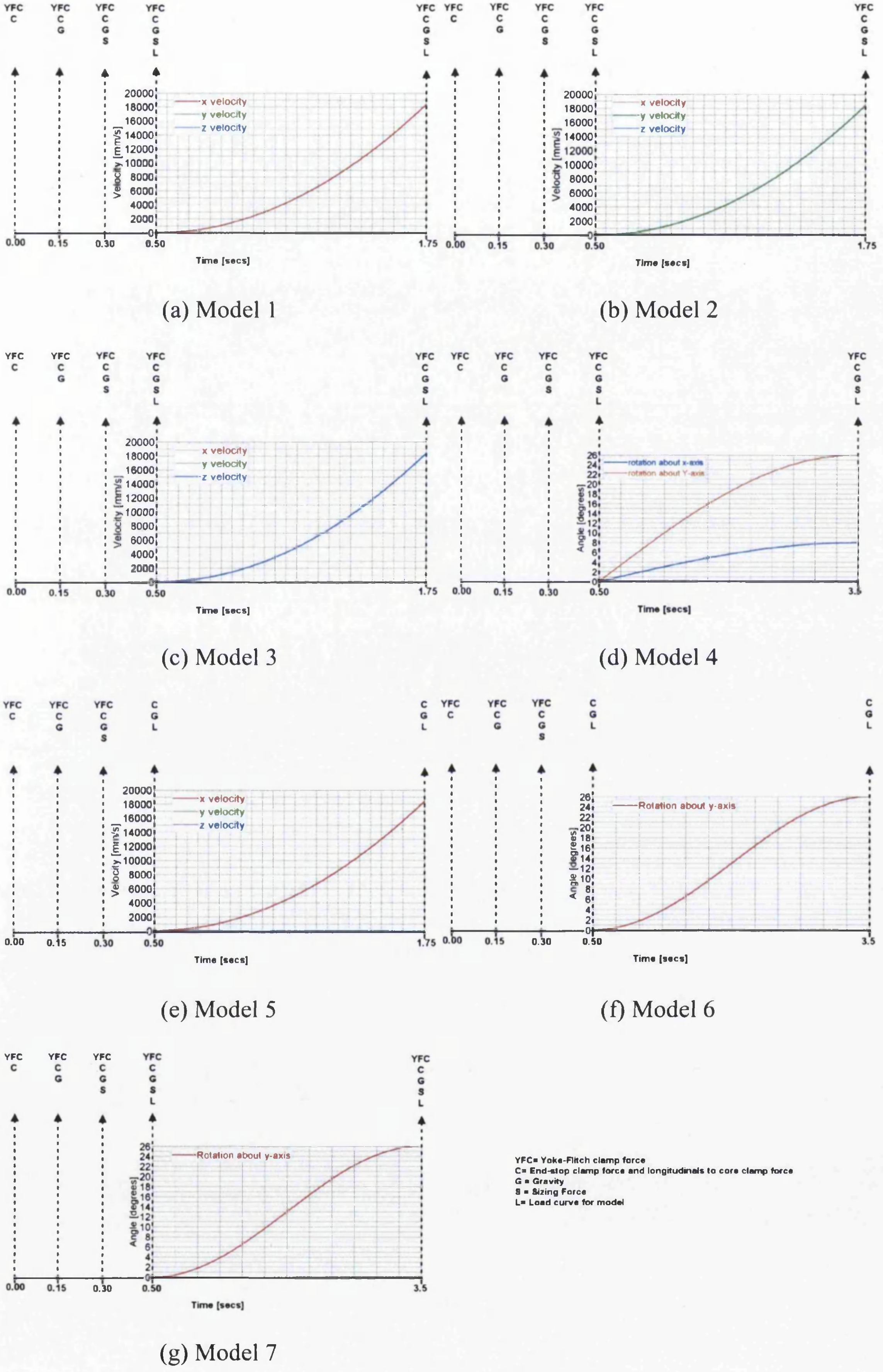


Figure 8.15: Relative motion of the yoke and flitch under cyclic accelerations in (a) negative x-direction, (b) no acceleration, (c) positive x-direction

Figures 8.16 to 8.22 show the loading time histories for the seven crash models in Table 8.1. Each loading history indicates at which point in time particular forces are applied and removed. It should be noted, that Models 1-4 and Model 7 maintain the clamping and sizing forces once they have been applied.



(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

(e) Model 5

(f) Model 6

(g) Model 7

Figure 8.16: Loading time histories for all of the models

8.6 COMPONENT INTERACTION AND LOAD PATHS

In this section the relative movement between different components and the sequence in which the movements occur is identified.

Figure 8.17 shows the stress distribution at various points in the assembly simulation (note the outer tank and the windings are not shown) when the following are applied:

(a) yoke_flitch, endstop, and longitudinals to core clamping forces.

(b) gravity loading.

(c) sizing force.

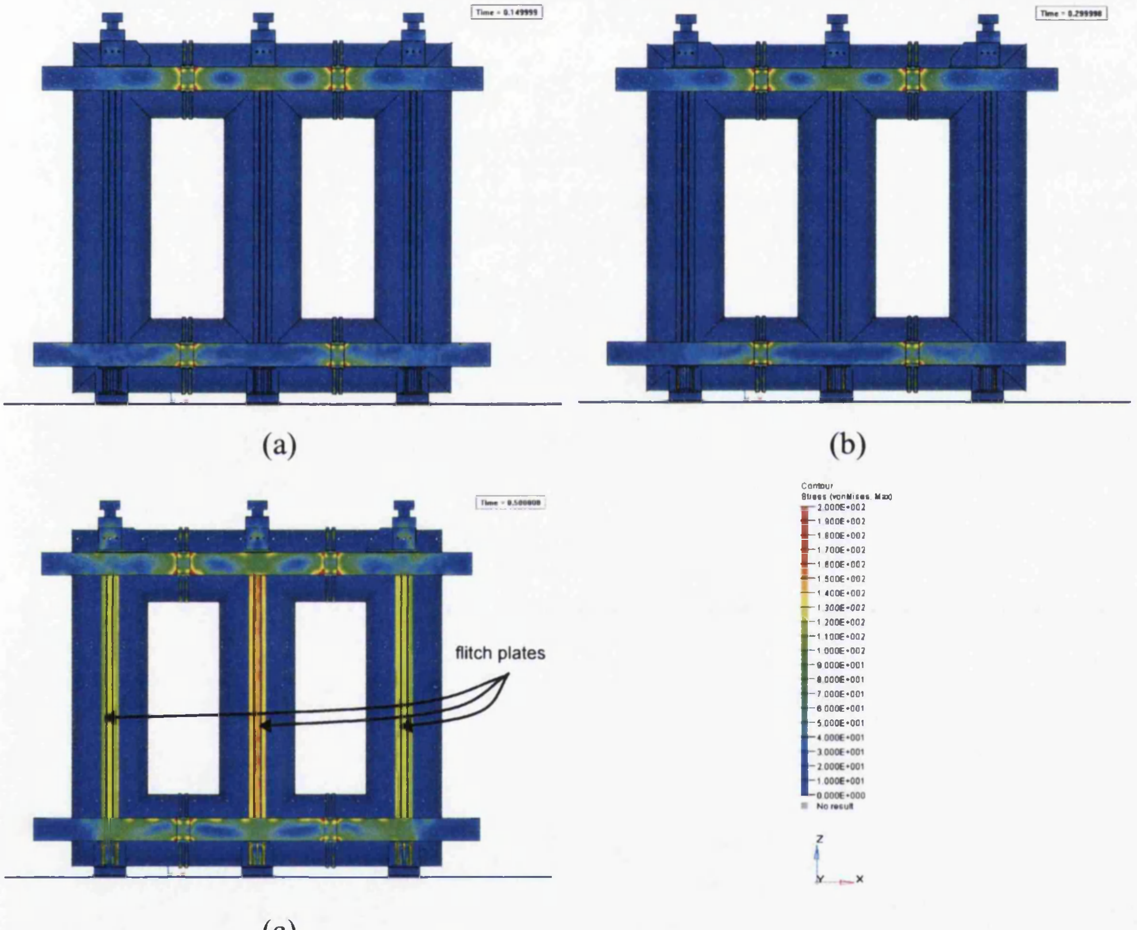


Figure 8.17: Stress distributions for the assembly simulation after application of (a) the yoke_flitch, endstop and longitudinals to core clamping forces, (b) the gravity loading and (c) the sizing force

It can be seen from Figure 8.17 that, when the sizing force is applied, the stresses in the flitch plates increase. This is to be expected, since the sizing force acts to tension the flitch plates.

There are four key areas where relative movement between the components affects the global behaviour of the model:

1. locate_bar to foot separation (Figure 8.18(a))
2. core to endstop separation (Figure 8.18(b))
3. core to baselock separation (Figure 8.18(c))
4. footpin to baseblock separation (Figure 8.18(d))

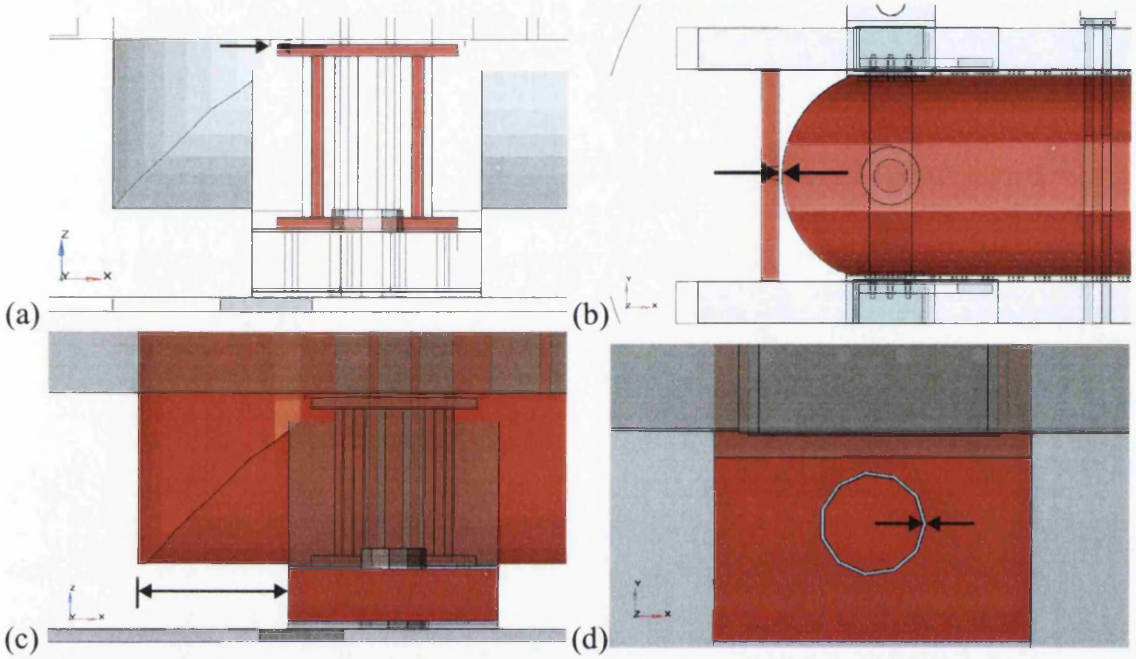


Figure 8.18: Key areas of relative movement: (a) locate_bar and foot; (b) core and top endstop; (c) core and baseblock; (d) footpin and baseblock

Acceleration in the y and in the +z -direction is uncritical up to 3.0g at which point the calculations were stopped. The typical movement sequence can be seen from the results of Model 1, where the movement is caused by the acceleration of the ship's deck in the x-direction. Figure 8.19 shows the relative position changes between the key areas identified above in Figure 8.18. It can be seen from Figure 8.19 that:

- At an acceleration of 0.1g, the core begins to move towards the top endstop, and the core to baseblock separation begins to decrease. Initially the rate of the position change is greater for the core to top endstop than the core to baseblock movement.
- At 0.38g the footpin to baseblock separation starts to decrease.
- At 0.5g the core and top endstop make contact.
- At 0.52g the footpin and baseblock make contact. At this point the rate of change of the relative movement of the core and baseblocks increases from approximately 8mm/s to 126mm/s.

- At 0.62g the locate_bar to foot separation starts to decrease and the entire transformer and the packaging, which is resting on the feet, begins to slide. At this acceleration, the yoke_flitch bolt von Mises stress is 128MPa.
- At 0.7g the foot hits the locate bar and the acceleration causes mechanical failure of the packaging.

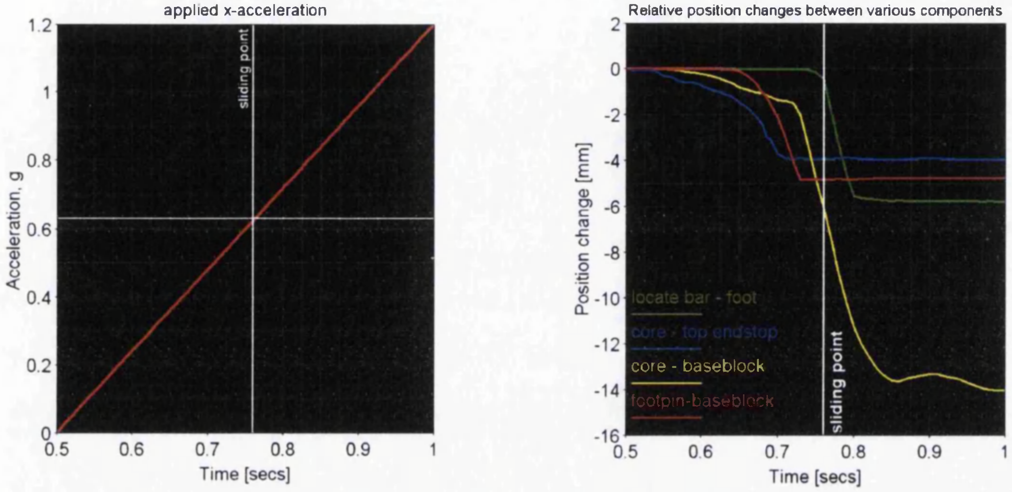


Figure 8.19: Model 1: Relative position changes between the key areas identified in Figure 8.18, caused by acceleration in the x-direction

As discussed earlier in Section 8.5, the yoke_flitch bolts are subjected to a cyclic loading. Assuming a cyclic loading of +128MPa/0MPa on top of the pre-tension loading of 362.5MPa, as calculated above, the bolts could loosen or even fail, and as a consequence, remove both the yoke_flitch clamping force and the sizing force. Figure 8.20, compares the separation curves, with and without these forces, i.e. the results of Model 1 and Model 5 respectively.

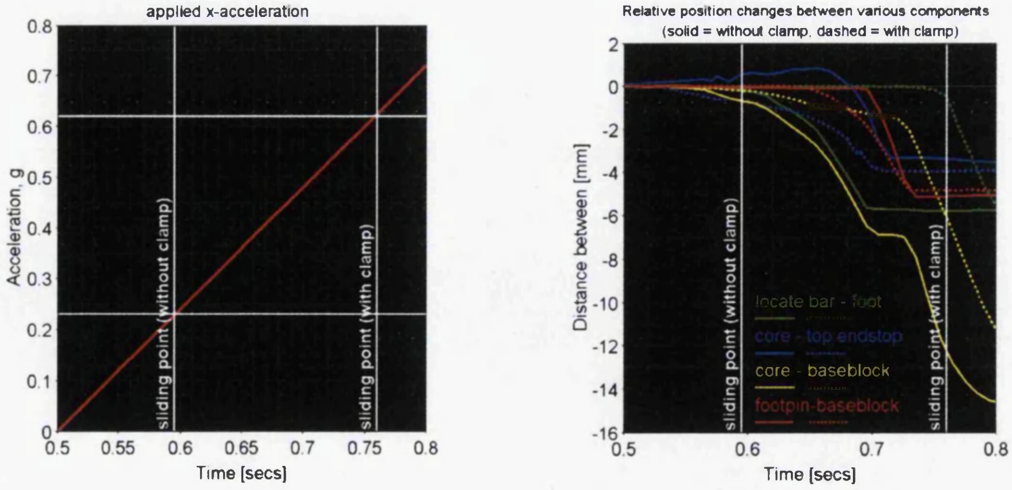


Figure 8.20: Model 1 and Model 5: Relative position changes between the key areas identified in Figure 8.18, caused by acceleration in the x-direction, with and without the yoke_flitch clamping force and the sizing force

The effects of loosening the yoke_flitch bolts can be seen in Figure 8.20 and are as follows:

- Sliding in the x-direction occurs at an acceleration of 0.23g compared to 0.62g.
- The core to top-endstop separation starts to decrease immediately.
- The core to baseblock separation starts to change at the same acceleration but the rate of change of position is 60mm/s compared to 8mm/s.
- The footpin to baseblock separation starts to change at a greater acceleration of 0.47g compared to 0.38g, but contact occurs at the same acceleration.

The effect of applying rotations to the model is that, the transformer starts to slide on the feet at a roll rotation angle of 12.6° with the yoke_flitch clamping force and sizing force applied (Model 7), this reduces to 9° when the forces are removed (Model 6).

8.7 CONCLUDING REMARKS

- In this chapter a computer model for a complex multi-body system subjected to a time-dependent, multi-axial loading has been developed to predict the behaviour of an electrical transformer and its protective packaging during sea transport. The study involved:
    - The identification of appropriate dynamic and static friction coefficients to conservatively model inter-component contact.
    - The formulation of the load curves to represent the motion of the sea.
    - The simulation of the loadings related to the assembly and securing of the transformer packaging.
    - The formulation of realistic loading regimes to simulate the sequence in which all of the loadings are applied in reality.
    - The identification of the interactions and the load path sequence for several different loading regimes.

- Overall, it can be seen that the effect of loosening the yoke_flitch bolts is that, the packaging and the transformer begin to move at an acceleration of 0.23g compared to 0.62g and the roll rotation angle reduces from 12.6° to 9°. However, even if the yoke_flitch bolts do not fail, the acceleration and rotation values are well within the likely maxima that occur in reality, so that, the reliance on friction to prevent the transformer from moving is not good and other mechanisms should be used to prevent the motion.

- Future work can be to address this issue, to investigate the effect of vibration on the design, and to look at the worst-case scenario of maximal roll and pitch angle combined with maximal accelerations. The FE model represents the "ideal" assembly conditions with consistent friction action over the contact areas, whereas "real" assembly is highly likely to introduce some inhomogeneities reducing the effect of friction and this is an area where further investigation is required.

8.8 References

[8.1] 'LS-DYNA: Non-linear Dynamic Analysis of Structures in Three Dimensions', LSTC, Version 960, 2002.

[8.2] Kaps H., Lüdicke A., "Safe stowage and storage of cargo", Bremen University of Applied Sciences, GAUSS institute, 2000.

[8.3] Beitz W., Grote K.H., "Dubbel - Taschenbuch für den Maschinenbau", 20th edition, Springer Verlag, Heidelberg, 2001. ISBN: 3-540-67777-1.

[8.4] Report No. QC99-135, "Coefficient of Friction Measurement for Non-oiled and Oiled surfaces", ETRS Pty Ltd., 28 April 1999.

[8.5] Personal communication with P. Giles, Gordon Giles & Company Ltd, consulting marine engineers & ship surveyors.

SECTION 3

CONCLUDING REMARKS

CHAPTER 9

CONCLUDING REMARKS

Summary: This chapter contains the achievements, the overall conclusions of the work and then recommendations for areas of future research.

9.1 ACHIEVEMENTS

The aim of this thesis was to advance the knowledge base of current computer-aided engineering (CAE) tools and to develop new CAE methods that address industrial design and manufacturing problems through the use of computer simulations and optimization. This section describes the resulting achievements.

Investigation of the slit die design problem in **Chapter 5** led to the following achievements:

(1) The performance of three optimization methods: one gradient-free, one gradient based and one hybrid method, have been compared in solving the deterministic shape optimization problem. **(See Section 5.4)**

(2) A procedure to solve RDO problems has been developed and implemented (see Chapter 4). The procedure has been used to solve the slit die design problem and comparison has been made between the deterministic optimum (found from the previous achievement) and the robust design. **(See Sections 4.4, 5.5, 5.6)**

(3) In order to solve the RDO problem a multi-objective optimization method is used to generate a Pareto set of efficient solutions. One method for generating the Pareto front has been developed. Three methods for generating a Pareto front are compared: NBI, PP and the new approach called the 'Pareto front marching method' using PP. **(See Section 3.5.4)**

(4) Whilst developing the robust design optimization procedure a general method for generating a space-filling design of experiments was required. A new method has been developed during this work for formulating the OLH DoE using the Audze-Eglais objective function. The formulation of this DoE is shown to be non-trivial. Two methods were developed, the first uses a binary encoded GA and the second uses a permutation encoded GA. The shortcomings of the former method are highlighted and the latter method was developed in response. **(See Section 3.4.2)**

The design of an A-pillar for Lea Francis Motor Ltd., described in **Chapter 6** led to the following achievements:

(1) A variety of CAE tools are brought together into a single design process, thus providing an initial feasible design without the need for prior physical prototyping and testing, and thereby reducing time and costs. Typical costs savings are a 20 to 30% decrease in the time to reach a manufacturable solution. The greatest savings, however, come from a reduction in the number of iterations to reach the solutions. Each iteration or refinement requires the manufacture and testing of the component. Using the approach implemented here, a reduction from up to 100 refinements to just one or two refinements can be achieved. The

process used to develop the final solution involves topology optimization, shape optimization, response surface approximation, manufacturing simulation and structural simulation.

(2) The design meets the FMVSS 216 roof crush requirements.

(3) A-pillar constructions are usually designed to be an extrusion and the novelty of the design is that it designed to be a cast component.

(4) Two approaches to reach the final design have been implemented. The first uses a combination of topology and shape optimization, the second uses topology optimization only, but with a finer mesh than the first approach.

Investigation of the seat comfort problem described in **Chapter 7** led to the following achievements:

(1) A review of the mechanisms that cause discomfort and some current CAE methods to improve comfort has been carried out. **(See Section 7.1)**

(2) A general procedure for the development of a CAE system for designing seats for comfort has been introduced. The system can be applied to the design of seats for single users and for multi-users. One part of the procedure is the novel idea of combining robust design and topology optimization to develop seat designs. **(See Section 7.2)**

(3) Investigations into the quantification and simulation of seat comfort have been carried out. **(See Section 7.3)**

The modelling of the behaviour of a cargo during sea transportation in **Chapter 8** led to the following achievements:

(1) A FE-based CAE model to predict the behaviour of a complex multi-body system subjected to a time-dependent, multi-axial loading has been developed. The model is used to predict the behaviour of a cargo and its protective packaging during sea transport.

(2) Identification of appropriate dynamic and static friction coefficients to conservatively model inter-component contact.

(3) The formulation of the load curves to represent the motion of the sea.

(4) The identification and the simulation of the loadings related to the assembly and securing of the cargo packaging.

(5) The formulation of realistic loading regimes to simulate the sequence in which all of the loadings are applied in reality.

(6) The identification of the interactions and the load path sequence for several different loading regimes.

9.2 CONCLUSIONS

In this section some conclusions are drawn. They are listed under appropriate (bold face) headings.

Designing under uncertainty

- ► As a result of the work carried out in Chapter 5 into RDO and from the literature, it can be concluded, that carrying out optimization under the assumption that there is no uncertainty in the system properties or boundary conditions doesn't always give the designer a good starting point. It can be the case that the optimal design is ideally optimal but really non-optimal, i.e. the deterministic optimum may not be optimum under real conditions although it is optimum under ideal conditions.
- ► There are many applications in the literature where optimization is used to solve stochastic problems by accounting for the uncertainty inherent in many design problems.
- ► From the work in Chapter 5 on the slit die design problem, it is observed, that if only one CPU is available, then the computational cost in terms of CPU time, of conventional deterministic optimization as opposed to stochastic RDO is proportional to the size of the sample used. If the sample size is 1000, then 1000 simulations are required for the stochastic optimization compared to one simulation for the deterministic optimization. Thus, the stochastic optimization would take a prohibitive amount of time. To address the problem, there are two possible solutions:
  - If the number of CPUs available equals the sample size then the samples can be all run in parallel meaning that the effective wall-clock time evaluation of performance is the same for both the deterministic optimization and the stochastic optimization. This situation is becoming increasingly feasible through the availability of cheaper and faster multi-processor machines.
  - If an accurate approximation of the design space incorporating the variation caused by the noise factors is available, then CPU time is no longer an issue.
- ► The development of the RDO method in Chapter 5 shows that the use of a small sample size enables rapid assessment and development of methods. The procedure is then used for a more realistic sample size that renders the problem computationally more expensive.

- ▶ The use of the small sample size in the development of the RDO method in Chapter 5, enabled rapid assessment of several multi-objective methods and a new method based called PFM-PP was developed.
- ▶ For the set-up of the methods in this work and for the current problem, PFM-PP generates a uniform distribution of points resulting in a smooth Pareto front. It also requires five-times fewer function evaluations to generate the solutions than the current NBI and PP implementations. PFM-PP can therefore be considered an efficient tool for generating robust designs.
- ▶ From a DoE study carried out on the noise factors in Chapter 5, it was shown that temperature has more influence on the mean and variation than the other noise variables meaning that the other noise factors had negligible impact on the response.

Multi-objective methods

- ▶ For the setup of the multi-objective methods described in Chapter 3 applied to the slit die problem in Chapter 5, the newly developed PFM-PP method generates a uniform distribution of points resulting in a smooth Pareto front. It also requires five-times fewer iterations to generate the solutions than the current NBI and PP implementations. PFM-PP can therefore be considered as an efficient tool for generating robust designs.

Optimization methods

- ▶ In Chapter 5 it can be seen, that there is a compromise between the quality of the results and the time taken to reach the result when choosing the appropriate optimizer. The GA finds better solutions than both DOT and MARS, but requires 10-times more function evaluations to reach the solution.

Design space approximations

- ▶ The advantages of using Monte Carlo simulations to solve problems of a stochastic nature are evident from the literature. However, it is also evident from the work carried out in Chapter 6 on the design of an a-pillar, that in a commercial environment there is competition for computer power and even with a simulation time of 20 minutes, it is only feasible to run a small number of simulations. In order to carry out optimization in a commercial environment it was necessary to use an approximation of the design space because the simulation times were too costly.

- From the literature, it can be seen, that the computational simulation can be anything representative of the problem's performance, e.g. an expensive solver, a low-fidelity model of the problem or a response surface model. In any case the choice of the type of simulation is and always will be dependent on the amount of computational power available as was the case in the a-pillar design in Chapter 6.

Design of experiments

- In developing the general method for formulating the OLH described in Chapter 3, it has been shown, that the permutation encoding (permGA) requires fewer function evaluations than the binary encoding and produces improved solutions to other existing solutions. The general method developed works for N variables and P points. It can be concluded, that the permGA method is an effective tool for developing OLH DoE.

Virtual design process

- From the work in Chapter 6 it can be seen, that bringing together a variety of CAE tools into a single design process provides the capability to reach an initial feasible design without the need for prior physical prototyping and testing.
- In Chapter 6 manufacturing simulations were used early in the design process to assess the a-pillar design. The results gave confidence that the design can be successfully manufactured and that time was not being 'wasted' on a design that cannot or was too expensive to be manufactured.

Topology optimization

- In Chapter 6 it was evident, that a design based purely on the topology optimization result compared to a solution that is modified according to engineering judgment and for manufacturing limitations does not necessarily perform better.
- When using the manufacturing constraints in the CAE tool OptiStruct [9.1], the results are dominated by the minimum member size constraint, which is enforced when using the manufacturing constraints.
- The use of a finer mesh requires considerable additional computer power, but it provides a cleaner design and removes ambiguities in the material distribution.
- The effect of using a finer mesh, and hence a smaller minimum member size is to generate a more complex and different material distribution.

Slit die design

- In Chapter 5 both deterministic optimization and RDO have been applied to the design of choker bars for use as a flow restricting mechanisms in a 1.2m wide, high-pressure slit die. The optimization has been used to determine the choker bar profiles required to give the best possible exit flow uniformity obtainable.
- For the particular die analyzed and from the deterministic optimization, increasingly sharp variations in the imposed restriction are required across the width of the die as the melt becomes more shear thinning. For the most extreme case, the choker bar profile fully spans its move limits, indicating that it is close to the limits of its effectiveness in bringing about an acceptably uniform flow distribution.
- Using optimization allows correct initial set up of the choker bar, reducing or eliminating the run-in time required for adjustment. Importantly, it also shows how effective the choker bar adjustment can be in a given application, and whether the best flow uniformity achievable is acceptable. Thus the practical operating range of a proposed die design, in terms of material properties and flow rates, can be determined in advance of practical trials.
- It will be necessary, in some cases at least, to machine a choker bar to the computed optimum profile, with the possibility of minor adjustments on-line by bending. This, in turn, suggests the use of interchangeable choker bars, optimized and machined for specific applications, thus providing the means to extend the operating range within which a die can produce uniform flow distribution.

Seat comfort design using CAE

- As a result of the work carried out in Chapter 7 it is evident, that the main target of designing seats for comfort is currently in the automotive and office industry and there is a need to implement CAE into the design process.
- Currently very little of the available technology has been implemented in wheelchair design and there is a huge market for cheap, easy to manufacture, comfortable (pressure reducing) seating, especially in developing countries, where pressure sores caused by inadequate wheelchair seating are the principal cause of death amongst disabled persons.

CAE models

- From the work carried out in Chapter 8, the development and use of a complex CAE model to predict the behaviour of a system can be used, not only for predicting deficiencies in a design before its use, but also for helping to establish why a system behaves in a certain way after an event.

9.3 FURTHER WORK

In this section further work is suggested under the appropriate (bold face) headings.

Robust design

- Determine the optimum values for the tolerances on the noise factors to bring about the minimum mean and variation of performance. This would also require factoring in the cost of meeting the tolerance specifications into the objective functions.
- Assuming that the tolerances are optimal then some of the noise factors can be eliminated from the study. In other words a DoE study is required initially to eliminate the insignificant noise factors. This is the approach used by Chen in [9.3]. Then optimization is required to determine the tolerances on the variation of the noise parameters with the constraints being the cost of tightening or relaxing them.
- Implement other techniques for robust design such as using response surface or kriging models and use the slit die problem as the 'test-bed', so easy comparison between the results can be made.
- A question arising from the comparison of the 10-sample results to the 100-sample results is: What sample size will give the best approximation of the mean and the variance of the population? Clearly the sample of size is 10 is an inadequate representation of the population and the sample size of 100 may also be inadequate. The appropriate size needs to be determined by further research investigating the results for other sample sizes. For this work the sample size is assumed to be appropriate.
- Further work into determining the appropriate distribution of the noise factors is required, although the assumption of a normal distribution is deemed satisfactory for the purposes of developing the RDO procedure.
- Identify factor significance prior to solving an RDO problem.
- There are two main options for the choice of approximation: firstly, create a global approximation of the entire design space and use this as the simulation code for all function evaluations. Or secondly, create an approximation of the local design space at each point specified by the optimizer. In this work local function evaluations are used and further work can be to look into the effect of using global or local approximations on the performance in RDO. In generating local approximations, only small changes in the design variables according to the tolerances are required, compared to global approximation where the full range of the design variable values are to be approximated.

Optimization method

- ▶ An alternative method to solve the deterministic problem maybe to use a GA to get to the optimum region of the design space and use a gradient-based optimizer to quickly zoom in on the optimum.

Multi-objective optimization

- ▶ Test PFM-PP for its ability to generate fewer or less points on Pareto front.
- ▶ Develop a method for generating the efficient frontier using a method such as that discussed in Chapter 4 by Zhang *et al.* [9.4].

Formulation of the optimal latin hypercube

- ▶ Improve the DoE encoding to decrease the number of design variables.
- ▶ Investigate the performance of GAs and the use of other discrete variable optimization algorithms to increase the efficiency of the method.
- ▶ The need to fully assess the quality of the results found using the permutation GA and the Audze-Eglais objective function. Using other objective functions, such as maximizing entropy, integrated mean-squared error, and the maximization of the minimum distance between points.

Topology optimization using manufacturing constraints

- ▶ Further work is required to identify the 'optimal' mesh size and minimum member size for which further reduction has no effect on the result or when the resulting material distribution is too complex to manufacture.

A-pillar design

- ▶ Introduce more shape variables in the lower section of the a-pillar.
- ▶ Determine the difference in the final a-pillar optimization results using different approximation methods.
- ▶ Carry out robust design for the a-pillar design. With the noise factors such as the angle, velocity and position that the loading plate hits the A-pillar, material variation e.g. yield stress and manufacture variation.
- ▶ Optimizing the position of the filling points to make the design more manufacturable.

FE-based model for predicting the behaviour of a cargo on a ship

- Future work can be to investigate the effect of vibration on the design, and to look at the worst-case scenario of maximal roll and pitch angle combined with maximal accelerations. The FE model represents the "ideal" assembly conditions with consistent friction action over the contact areas, whereas "real" assembly is highly likely to introduce some inhomogeneities reducing the effect of friction and this is an area where further investigation is required.

The worst-case scenario consisting of maximum roll and pitch overlaid with maximum accelerations (acceleration is maximal when ship is rotated by maximal angles) could be analysed. The computation would require a full assembly simulation, followed by a full rotation analysis, followed by a quarter period acceleration at the maximum roll and pitch angles.

9.4 FURTHER THOUGHTS

- The approximation of design spaces is a very important area of research. Owing to time constraints and limitations on CPU time availability the competition for computational time in industry is acute. As computers are becoming increasingly powerful the simulations are becoming more and more complex. Until such time as the physics of a system are fully captured in the simulations, CPU constraints will remain an issue.
- The best solution is only the best for the model being used and so the limitation of a design optimization process is the complexity of the model. Therefore, the more the physics can be captured in a model the more likely it will represent the true behaviour and the design space will contain the true optimum.
- In stochastic optimization many thousands of function evaluations are required and so approximations are used to replace the expensive solver. The main advantage of an approximation is that many approximate simulations can be carried out in the time it takes to do one expensive simulation. In Section 3.3 it was shown that using ANN generates a more accurate approximation than a PA but required 10-times more function evaluations. In general, this is the case, that the more accurate the approximation the more function evaluations are required.
- "In systems dominated and characterized by a mixture of chaotic and random behaviour, talking of optimization is simply absurd. You can't optimize a car for

crash when you don't know what other car, or obstacle, it will hit, with which velocity, angle, mass, number of passengers etc. Because of tolerances in manufacturing and assembly, it is even impossible to manufacture two identical cars. So what can we optimize?" [9.2] In answer to the question, what can be done is to develop and to modify the safety requirements of the designs as the technologies of modelling and CPU power improve. The requirements must remain obligatory for the safe design of cars, but they must also be updated to match advances in technology. In this way the requirements become more sophisticated as the technology becomes more sophisticated, what remains constant is that the design must be optimized to meet the safety requirements.

- ► Currently there are number of safety requirements which a car design must pass, e.g. the roof crush requirement described in Chapter 6, the nature of the test may be insufficient to cope with all roof crush eventualities but it is appropriate to the current technology available. If it were more complex, then it would be almost impossible to come up with a car design that can pass a test. As techniques in stochastic optimization become more computationally efficient and designers are able to model variation in the system parameters efficiently, the more complex the safety requirements can become and the closer these requirements will match all possible real world scenarios.

9.5 References

[9.1] "Altair OptiStruct Version 6.0', Altair Engineering Limited, 2003.

[9.2] Marcyk J. "Automotive crash simulation: a personal perspective" Easi Engineering, January 2001.

[9.3] Chen, W. "Robust Conceptual Exploration for Configuring Complex Engineering Systems," Ph.D. Dissertation, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, August 1995.

[9.4] Zhang J., Wiecek M.M., Chen W., "Local approximation of the efficient frontier in robust design", 1999 ASME design engineering technical conferences, Las Vegas, Nevada. DETC99/DAC-8566, September 12-15, 1999.

••••••••••••••••••••••

APPENDICES

APPENDIX A1

OPTIMIZATION MODELLING

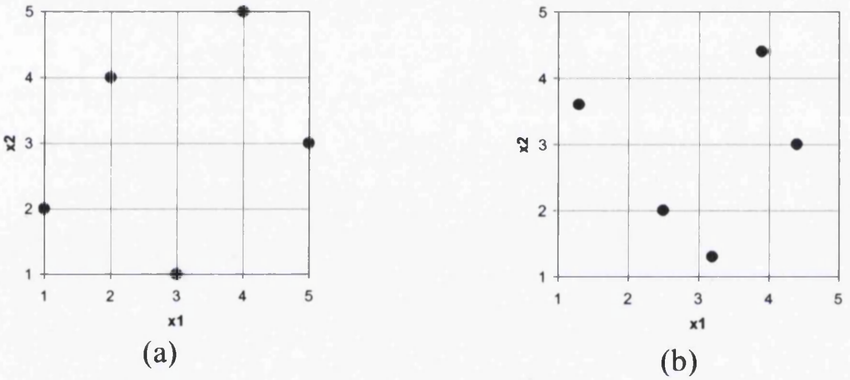
In this appendix further results for the OLH formulation discussed in Chapter 3 are given. Then, an example of how the physical programming method is provided.

A1.1 Optimal Latin Hypercube DoE: further results

This section provides some further results for formulating the optimal Latin Hypercube DoE using the Audze-Eglais objective function (AELH) discussed in Chapter 3.

DoE for 5 points and 2 design variables

The DoE for 5 points and 2 design variables is shown in Figure A1.1. Figure A1.1(a) shows the AELH DoE formulated using the GA. Figure A1.1(b) shows the RLH DoE generated using [3.96]. Inspection of Figure A1.1 indicates the AELH DoE has a better uniformity than the RLH DoE does. Potential energy comparison, using Table A1.1, backs this up as the AELH has a lower potential energy than the RLH DoE does.



(a)

(b)

Figure A1.1: Plan points for *P*=5 and *N*=2; (a) AELH using binGA, (b) RLH

	2 × 5	2 × 10	2× 120	3 × 5	3 × 10	3 × 120	5 × 50	5 × 120	8×99
AELH binGA	1.30	2.07	5.77	0.73	1.04	2.05	0.73	0.80	0.42
AELH –results from [3.97]	–	2.11	–	–	–	–	0.73	–	–
RLH from [3.96]	2.75	4.08	10.44	2.11	2.30	2.64	0.88	0.89	–

Table A1.1: Potential energy comparison of RLH designs with AELH binGA designs

DoE for 10 points and 2 design variables

The DoE for 10 points and 2 design variables is shown in Figure A1.2. Figure A1.2(a) shows the AELH DoE formulated using the GA. Figure A1.2(b) shows the AELH DoE given in [3.97]. Figure A1.2(c) shows the RLH DoE generated using [3.96]. Potential energy comparison between Figure A1.1(c) and A1.1(a) given in Table A1.2 shows that the GA DoE formulation has a lower potential energy than a previous solution to the problem taken from [3.97]. This indicates that the method introduced in this study works and that the GA finds an improved solution.

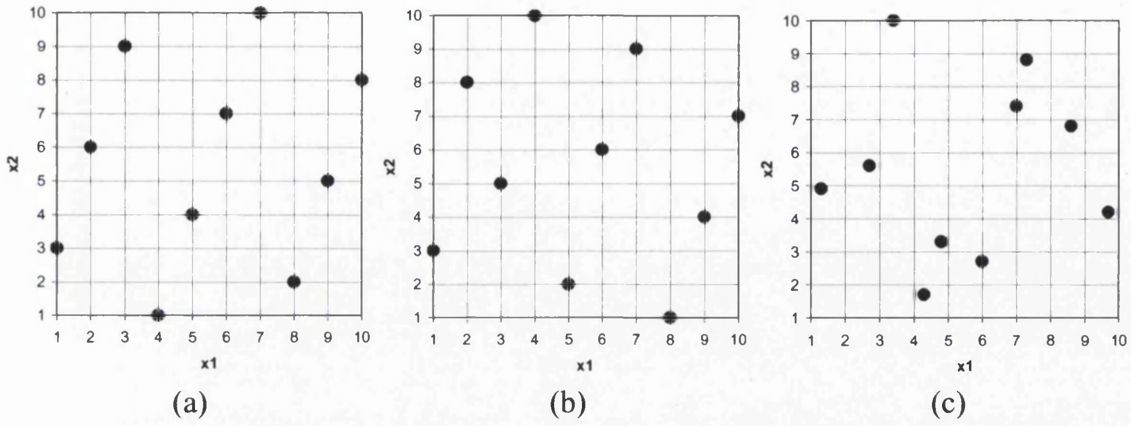


Figure A1.2: Plan points for *P*=10 and *N*=2; (a) AELH using binGA, (b) AELH from [3.96], (c) RLH

DoE for 5 points and 3 design variables

The DoE for 5 points and 3 design variables is shown in Figure A1.3. Figure A1.3(a) shows the AELH DoE formulated using the GA. Figure A1.3(b) shows the RLH DoE generated using [3.96]. Inspection of Figure A1.3 indicates that the AELH DoE has a better uniformity than the RLH DoE does. Potential energy comparison, using Table A1.1, backs this up as the AELH has a lower potential energy than the RLH DoE does.

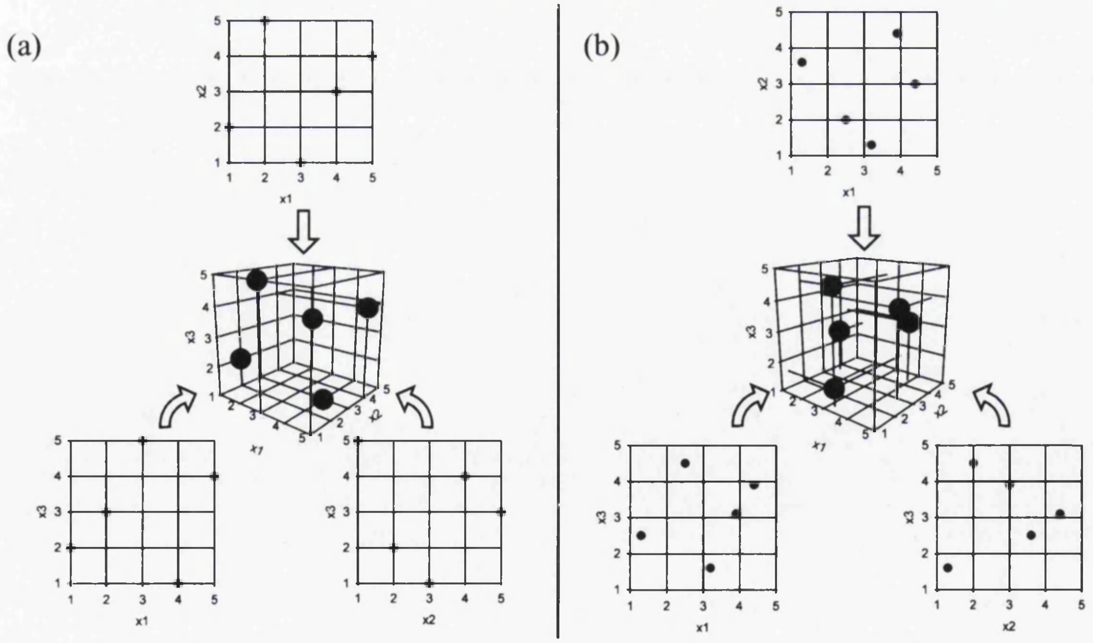


Figure A1.3: (a) AELH using binGA; (b) RLH, for P=5 and N=3; 3D-view(middle) and corresponding plan view $x_1 \times x_2$ (top), side view $x_1 \times x_3$ (bottom-left) and side view $x_2 \times x_3$ (bottom-right)

DoE for 10 points and 3 design variables

The DoE for 10 points and 3 design variables is shown in Figure A1.4. Figure A1.4(a) shows the AELH DoE formulated using the GA. Figure A1.4(b) shows the RLH DoE generated using [3.96]. Inspection of Figure A1.4 indicates that the AELH DoE has a better uniformity than the RLH DoE does. Potential energy comparison, using Table A1.1, backs this up as the AELH has a lower potential energy than the RLH DoE does.

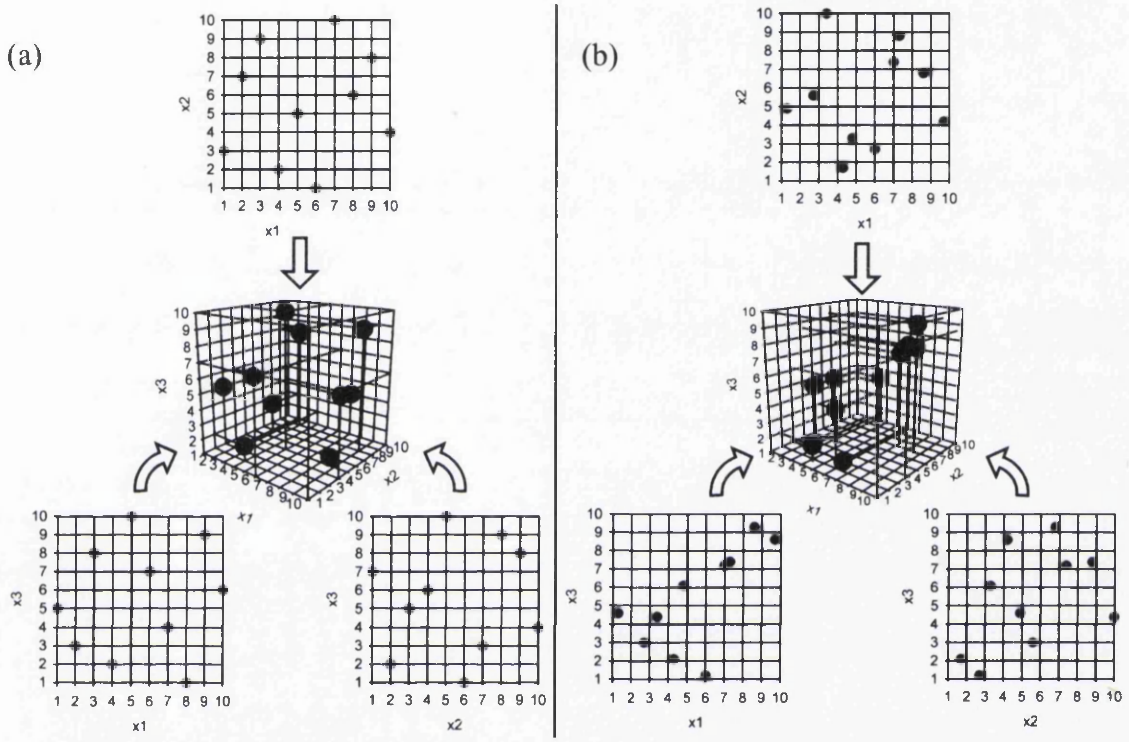


Figure A1.4: (a) AELH using binGA; (b) RLH, for P=10 and N=3; 3D-view(middle) and corresponding plan view $x_1 \times x_2$ (top), side view $x_1 \times x_3$ (bottom-left) and side view $x_2 \times x_3$ (bottom-right)

DoE for 120 points and 3 design variables

The DoE for 120 points and 3 design variables is shown in Figures A1.5. Figure A1.5(a) shows the AELH DoE formulated using the GA. Figure A1.5(b) shows the RLH DoE generated using [3.96]. Inspection of Figure A1.5 indicates that the AELH DoE has a better uniformity than the RLH DoE does. Potential energy comparison, using Table A1.1, backs this up as the AELH has a lower potential energy than the RLH DoE does.

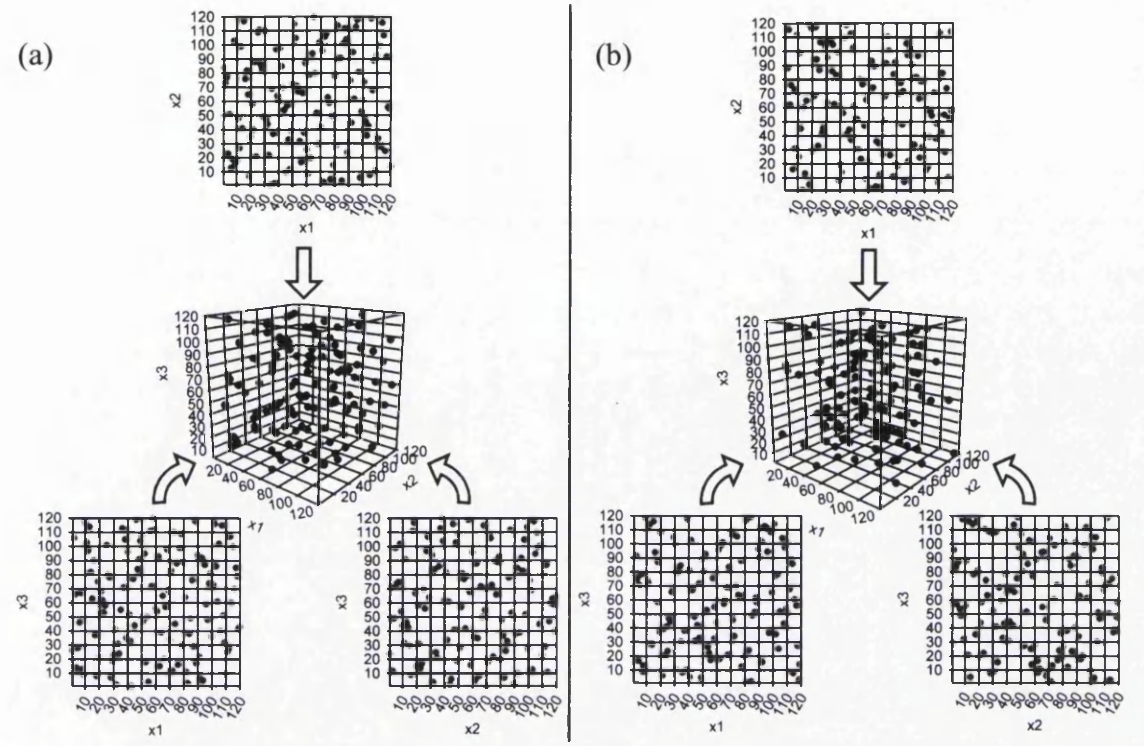


Figure A1.5: (a) AELH using GA plan points; (b) RLH plan points, for P=120 and N=3; 3D-view(middle) and corresponding plan view $x_1 \times x_2$ (top), side view $x_1 \times x_3$ (bottom-left) and side view $x_2 \times x_3$ (bottom-right)

DoE for 50 points and 5 design variables

The DoE for 50 points and 5 design variables is given in Table A1.2. The previous AELH DoE is taken from [3.97] and the RLH DoE is generated using [3.96]. Potential energy comparison between the previous solution and the GA solution shows that the two DoE have the same potential energy. This again indicates that the GA method has worked as expected.

DoE for 120 points and 5 design variables

The AELH DoE for 120 points and 5 design variables is given in Table A1.3 generated using binGA. The RLH DoE is generated using [3.96]. Potential energy comparison, using Table A1.1, indicates that the AELH DoE has a better uniformity than the RLH DoE because the AELH DoE has a lower potential energy than the RLH DoE.

AELH – GA results					AELH –results from [7]					RLH				
x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5
1	36	22	24	43	1	45	19	23	37	1.3	32.2	40.7	34.5	32.9
2	17	34	43	23	3	40	23	38	14	2.9	15.1	7.5	14.4	21.9
3	9	39	18	40	2	15	43	19	30	3.6	28.6	16.1	7	43.9
4	10	40	15	17	4	43	22	2	26	4.6	18.2	15.2	45.7	10.9
5	6	12	14	25	6	26	32	28	2	5.2	25.5	29.8	31	38.9
6	31	9	27	19	5	24	44	47	23	6.5	13.3	4.3	22.2	12.7
7	20	28	30	1	7	21	28	5	12	7.5	44	25.2	8.5	34.9
8	41	43	32	16	8	32	31	42	44	7.9	7.4	9.1	39.8	46.5
9	32	41	5	21	9	20	9	49	27	9.3	24.2	33.9	46.5	40.9
10	14	11	48	18	11	22	5	26	7	10.5	38.1	41.4	10.4	25.7
11	49	18	37	9	10	4	26	44	22	11.2	35.5	29.2	48.2	41.6
12	2	20	36	39	12	42	46	14	29	12.1	17.3	45.8	39.1	1.1
13	26	3	21	47	14	35	3	35	42	13.3	5	38.7	29.6	33.8
14	38	45	45	42	13	16	12	21	49	14	16.1	17.8	8.9	45.8
15	23	5	13	7	16	48	39	34	13	14.9	1.7	3.1	28.8	14.4
16	28	46	12	46	15	1	18	9	28	15.9	26.4	36.1	33.4	17.4
17	50	31	19	41	17	9	37	16	45	17.6	8.7	23.2	13.1	23
18	45	13	8	27	18	10	50	36	34	18	5.9	43.3	31.8	8.4
19	39	1	39	38	19	36	6	7	36	19.5	49.5	19.6	27	37.5
20	18	29	47	48	20	5	40	31	6	20	28.4	28.2	44	13.5
21	15	50	34	13	22	47	27	13	5	21.4	39.7	44.6	6.2	49.7
22	44	19	44	31	21	6	15	37	43	22.4	45.3	13.7	40.8	39.7
23	33	2	40	8	23	41	33	11	48	23.1	19.6	10.8	16.4	10.3
24	30	23	3	49	24	25	48	6	15	24.1	33.7	46.3	20.6	19.5
25	37	35	38	2	25	37	7	48	19	24.9	44.9	38.2	48	27.9
26	13	24	22	50	26	18	14	1	40	25.6	30.8	48.5	44.8	20.4
27	1	17	25	11	27	33	47	33	47	26.9	26.8	6	32.6	37.1
28	35	38	9	4	28	50	8	24	33	27.6	36.8	14.6	5.6	30.4
29	47	21	17	6	30	34	2	10	11	29.1	12.7	24.2	37.5	47.7
30	43	36	2	28	29	7	1	18	20	30.4	14.7	32.3	17	17.8
31	16	49	42	35	31	19	35	50	39	30.4	22	32.8	14.7	23.9
32	24	8	49	24	32	8	24	12	1	31.7	9.1	26.3	26.1	27.1
33	5	48	11	20	33	27	41	46	8	33	11.4	21.2	24.9	9.5
34	11	15	7	5	34	17	13	41	3	34	2.4	31.1	19.1	48.8
35	46	42	41	32	35	11	4	40	21	34.4	48	27.1	28.3	5.5
36	8	25	4	36	36	23	42	4	35	35.4	30	2	21.3	16.4
37	25	7	6	34	37	2	45	27	25	36.7	46.8	40.1	24.1	4.7
38	21	32	1	12	38	28	10	29	50	37.5	38.4	5.7	4.4	35.3
39	4	16	46	29	39	31	25	3	10	38.7	3	12.6	35.7	25.3
40	22	37	50	15	41	39	16	32	4	40	47.8	11.5	2.9	15.1
41	34	4	16	10	40	38	20	45	38	40.4	40.5	37.2	41.9	7.4
42	42	33	23	44	42	3	17	8	24	41.4	20.6	35.2	23.5	31.4
43	19	47	20	37	43	29	49	25	16	42.4	32.5	19.8	49.8	44.4
44	12	27	33	3	44	44	36	20	41	43.8	19.4	49.7	3.1	2.7
45	40	44	26	14	46	49	30	17	17	44.1	35.2	1.1	11.2	30
46	29	26	35	22	45	46	38	39	18	45.8	4.2	8.6	42.8	43.1
47	7	6	10	26	47	12	29	30	9	46.9	10.3	47.4	18	5.9
48	3	30	29	33	48	13	21	43	31	47.2	42.9	22.2	37	21.2
49	27	10	31	45	49	14	34	22	46	48.5	22.9	43	12.3	28.8
50	48	14	28	30	50	30	11	15	32	49.2	41.7	17	1.8	3.7

Table A1.2: DoE co-ordinates for 50 points and five design variables

AELH- GA results										RLH									
x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5
1	86	80	86	79	61	119	61	54	110	1.3	88.6	40.3	37.5	73.6	61.3	2.3	95.8	28.0	115.6
2	57	65	72	36	62	58	87	34	120	2.9	88.2	91.7	77.1	33.5	62.4	93.3	18.0	65.8	56.1
3	75	41	118	75	63	84	76	7	11	3.7	115.6	75.4	82.7	40.8	63.4	74.4	99.1	39.6	77.2
4	73	91	8	61	64	1	84	70	46	4.6	62.4	16.7	59.4	7.3	64.4	89.5	101.1	25.5	45.8
5	26	59	35	20	65	111	6	45	53	5.2	76.1	70.8	72.7	67.5	65.2	4.1	57.2	93.7	47.3
6	20	103	69	38	66	27	9	100	69	6.5	16.3	78.4	92.7	97.6	65.8	36.2	59.9	34.6	39.8
7	83	7	64	57	67	33	115	76	10	7.6	30.9	8.9	114.6	79.1	67.4	34.8	68.9	118.7	52.6
8	14	72	37	94	68	47	109	104	44	7.9	72.8	13.4	56.8	83.8	68.0	12.5	21.0	106.0	15.4
9	45	62	95	114	69	50	5	22	78	9.4	111.9	118.5	87.6	104.1	68.8	3.8	83.6	61.5	104.9
10	70	30	99	12	70	2	18	65	77	10.6	25.2	74.3	5.7	112.1	70.1	26.8	109.2	3.0	10.5
11	43	11	39	95	71	55	66	115	5	11.3	99.0	72.6	63.8	30.8	71.0	50.3	106.7	79.4	91.6
12	35	13	84	56	72	60	17	111	103	12.2	45.2	113.7	18.5	32.1	71.7	68.1	45.7	79.2	64.6
13	59	94	78	4	73	102	112	87	55	13.5	1.5	29.5	53.5	107.6	72.7	91.7	34.3	53.7	115.0
14	67	93	14	105	74	89	92	79	2	14.1	60.2	28.0	33.1	38.7	73.8	110.5	69.9	89.4	44.2
15	114	97	47	88	75	16	74	13	48	15.1	13.3	116.0	108.4	17.1	74.5	100.8	105.2	88.5	66.6
16	54	33	31	42	76	11	47	91	115	16.1	65.1	94.1	101.0	102.4	75.4	26.2	63.6	112.3	113.9
17	116	34	23	62	77	80	120	55	26	17.8	119.4	58.7	13.3	6.1	77.4	11.0	88.0	38.3	77.4
18	24	79	81	81	78	9	26	15	50	18.2	5.4	87.1	47.8	58.8	77.9	83.9	12.6	107.4	22.4
19	100	106	114	63	79	113	101	44	68	19.7	116.5	37.1	45.7	60.2	79.0	92.4	63.2	44.6	37.2
20	74	32	58	104	80	63	70	53	24	20.2	29.2	107.9	90.3	71.5	79.5	20.0	19.7	68.1	27.1
21	38	110	102	45	81	31	68	117	43	21.6	68.5	6.1	23.0	19.1	80.6	18.5	116.3	85.1	50.0
22	110	77	48	32	82	17	56	32	109	22.7	117.4	39.4	70.4	9.0	81.6	70.0	1.4	41.1	35.9
23	109	22	96	47	83	93	52	26	118	23.4	94.4	20.5	103.2	43.0	83.1	77.7	24.2	3.8	116.7
24	107	37	88	106	84	79	31	66	117	24.4	33.0	41.1	9.3	58.3	84.2	48.1	119.0	55.4	79.7
25	5	28	36	72	85	51	85	2	92	25.2	86.8	23.7	21.6	13.5	84.7	71.9	2.0	15.0	21.2
26	98	46	24	98	86	99	64	119	30	25.9	106.6	46.9	101.9	100.8	85.5	7.2	48.8	60.4	98.9
27	48	117	28	74	87	77	114	101	101	27.2	42.3	44.8	6.3	105.7	87.1	105.7	51.4	35.8	61.3
28	53	27	116	51	88	115	73	19	28	27.9	45.6	65.8	63.0	81.0	87.8	97.5	61.6	110.0	24.6
29	97	116	71	54	89	25	107	21	29	29.4	104.0	104.9	14.4	1.1	88.3	22.1	58.5	107.1	38.2
30	4	49	67	15	90	88	16	17	90	30.7	63.9	11.1	104.8	96.3	89.7	101.9	96.3	49.8	82.8
31	46	104	33	31	91	117	39	82	37	30.8	95.7	99.8	113.1	26.0	90.6	90.9	90.5	25.8	119.6
32	15	12	46	23	92	56	60	105	108	32.1	108.3	79.7	86.9	95.6	92.1	81.0	101.8	75.3	70.2
33	66	45	38	1	93	68	35	56	3	33.4	108.1	83.2	24.5	18.3	92.7	33.8	15.7	83.3	107.0
34	64	111	107	86	94	65	98	20	13	34.4	85.9	84.7	31.5	117.8	94.1	82.0	112.6	32.7	75.5
35	69	71	41	85	95	21	100	43	102	34.8	104.9	30.4	30.6	87.7	95.2	96.9	112.0	42.0	4.1
36	7	50	62	111	96	82	29	10	22	35.8	62.6	8.2	71.4	94.6	96.1	66.5	94.4	8.1	85.0
37	71	1	49	35	97	91	118	25	96	37.1	83.3	38.1	116.1	68.7	96.4	24.6	110.6	74.2	25.2
38	18	88	98	16	98	23	3	77	41	38.0	99.8	10.0	40.4	90.1	97.2	78.8	36.5	7.4	91.2
39	62	15	4	18	99	103	78	9	80	39.2	8.3	117.7	69.4	99.8	98.4	32.2	108.3	29.3	101.2
40	52	105	68	113	100	81	82	74	119	40.4	19.5	81.6	85.3	74.8	99.4	40.2	65.3	80.7	118.0
41	39	36	6	100	101	92	8	106	70	40.9	14.2	5.2	1.3	110.8	100.9	15.7	114.2	51.5	66.1
42	19	38	112	25	102	40	19	29	107	41.9	56.2	102.6	71.8	16.8	101.6	58.8	77.1	94.7	8.0
43	32	58	3	39	103	112	53	50	60	42.9	109.9	67.1	35.7	54.8	102.2	17.7	35.5	115.9	81.4
44	104	83	83	112	104	10	86	57	19	44.4	118.9	48.0	60.7	23.6	103.4	39.6	78.3	49.3	109.3
45	6	95	18	89	105	28	42	90	21	44.7	41.5	31.0	26.9	49.3	104.2	57.4	25.2	110.7	61.6
46	87	67	1	67	106	36	44	12	14	46.3	44.6	42.2	81.8	20.4	105.1	54.5	51.8	42.8	28.3
47	118	69	109	59	107	72	24	94	33	47.4	112.8	71.8	4.3	34.7	106.6	70.8	104.1	11.0	93.0
48	101	54	103	6	108	105	51	97	97	47.7	37.3	32.9	47.1	70.9	107.9	38.3	43.2	98.2	86.6
49	106	108	27	17	109	30	102	59	65	49.1	102.8	4.8	99.0	35.6	108.4	48.9	97.4	52.4	47.7
50	42	4	80	7	110	8	75	85	93	49.8	61.3	89.0	119.4	51.7	109.6	6.6	53.8	102.9	108.3
51	41	10	73	116	111	94	99	93	27	50.7	51.6	80.6	12.6	53.8	110.7	113.5	50.6	10.4	30.3
52	96	20	5	49	112	13	43	61	8	52.0	51.9	92.8	16.0	14.3	112.0	42.8	67.6	92.2	11.5
53	22	48	113	99	113	29	55	11	76	53.1	30.3	52.8	97.2	29.5	112.7	84.7	89.4	78.0	3.7
54	49	81	120	84	114	37	40	63	82	53.9	80.3	26.6	118.0	45.4	114.0	55.2	85.8	57.5	5.5
55	120	63	60	34	115	78	57	92	71	54.7	23.7	22.4	95.3	64.4	114.5	28.5	7.9	67.0	89.0
56	85	113	16	58	116	95	21	30	64	55.8	76.7	17.1	45.2	94.1	115.5	74.0	60.6	112.1	51.1
57	3	89	89	87	117	44	14	40	40	57.2	47.4	75.5	17.6	12.0	116.7	10.6	56.0	22.5	57.2
58	12	119	42	66	118	76	96	51	91	58.3	9.4	32.4	56.1	86.0	117.5	52.7	27.3	65.4	112.6
59	108	25	52	9	119	34	23	110	73	58.5	65.7	55.0	76.2	73.2	118.7	114.2	3.5	20.2	63.2
60	90	2	75	83	120	61	90	108	52	60.0	21.4	43.9	19.2	2.6	119.0	58.0	14.2	99.3	42.5

Table A1.3: DoE co-ordinates for 120 points and 5 design variables

DoE FOR 99 POINTS AND 8 DESIGN VARIABLES

Table A1.4 shows the uniformly distributed plan points for AELH DoE for 99 points (plus one nominal point) and 8 design variables, generated using the binGA. Table A1.5 shows the corresponding normally distributed plan points generated using the technique discussed in Section 4.2.1.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
50	50	50	50	50	50	50	50
1	71	49	54	49	22	26	7
2	78	24	65	15	26	74	70
3	4	65	35	79	20	34	35
4	44	15	68	81	25	57	80
5	49	23	69	63	97	77	47
6	52	58	52	80	8	94	28
7	13	74	13	23	77	70	57
8	89	96	29	40	70	62	37
9	15	21	57	13	38	37	15
10	64	13	92	21	23	29	50
11	56	71	2	72	14	46	56
12	70	91	74	48	3	58	62
13	54	88	39	42	94	16	23
14	69	45	27	11	13	68	20
15	16	47	77	31	10	18	72
16	67	10	42	12	65	10	49
17	24	46	41	90	82	35	94
18	29	55	83	84	48	98	73
19	46	98	81	44	64	17	60
20	88	20	89	58	50	43	6
21	86	77	80	69	89	67	75
22	60	48	4	85	92	53	32
23	1	26	7	43	57	72	46
24	51	75	91	97	67	32	17
25	81	42	20	50	74	2	74
26	36	29	94	41	73	24	96
27	72	76	85	5	78	41	16
28	95	82	40	96	46	52	21
29	7	2	56	54	37	88	76
30	37	81	21	34	83	12	97
31	85	7	31	71	31	22	83
32	21	64	45	9	42	80	95
33	59	93	73	59	72	97	12
34	25	61	5	24	12	30	88
35	8	80	86	55	27	78	26
36	33	40	82	10	95	81	31
37	97	32	78	95	45	83	44
38	12	17	95	83	34	40	30
39	47	11	6	27	84	75	27
40	2	69	44	66	81	51	3
41	76	30	15	77	33	4	18
42	6	78	96	19	61	69	65
43	9	36	12	36	39	15	11
44	58	90	71	2	41	96	45

Table A1.4: AELH DoE co-ordinates for 99 points (plus one nominal point) and 8 design variables uniformly distributed

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
45	18	89	55	78	29	11	85
46	62	35	16	74	28	92	93
47	14	8	49	86	91	61	40
48	75	72	9	82	71	86	78
49	32	3	50	37	98	20	22
50	28	59	11	52	2	95	41
51	91	9	25	22	49	93	68
52	35	6	19	7	36	33	91
53	92	79	38	17	7	27	55
54	23	97	46	88	96	59	67
55	39	28	72	1	17	50	8
56	79	43	84	53	11	82	92
57	84	37	30	3	18	64	86
58	17	56	14	51	53	87	2
59	99	44	1	26	62	54	38
60	77	16	67	61	9	71	4
61	73	94	60	25	60	76	99
62	83	4	24	92	58	48	19
63	96	57	64	29	90	6	48
64	3	41	3	62	56	39	84
65	94	54	61	94	4	31	64
66	61	1	10	73	86	56	71
67	98	68	48	56	15	90	24
68	38	85	23	70	1	21	34
69	34	95	34	87	63	84	29
70	41	31	8	14	80	14	25
71	93	14	98	47	47	36	61
72	55	83	87	99	52	23	58
73	19	19	33	89	5	66	63
74	50	51	75	75	75	99	77
75	20	66	79	4	32	9	33
76	53	99	32	33	51	1	51
77	65	63	97	45	59	7	87
78	57	86	53	32	21	63	1
79	11	87	58	16	76	25	79
80	27	39	66	60	19	3	90
81	82	25	88	64	88	38	13
82	63	52	63	91	93	8	52
83	90	60	28	76	43	19	89
84	68	92	17	28	87	44	69
85	43	18	43	6	79	91	66
86	74	70	99	8	55	49	53
87	87	67	22	65	69	89	36
88	26	34	37	98	40	47	14
89	42	12	76	46	6	13	43
90	5	27	90	57	68	42	39
91	80	33	47	18	66	73	5
92	31	73	18	30	85	45	10
93	48	50	70	67	44	5	9
94	22	62	93	68	24	79	54
95	66	22	26	20	16	28	42
96	30	38	59	39	99	55	81
97	45	53	36	93	54	60	98
98	40	5	62	35	30	65	82
99	10	84	51	38	35	85	59

Table A1.4: ***(continued)***

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
50	50	50	50	50	50	50	50
1	59.1371	49.5866	51.6565	49.5866	37.2345	39.3729	25.3452
14.3093	62.7655	38.3288	56.3583	32.8252	39.3729	60.6271	58.6575
17.9518	20.43	56.3583	43.6417	63.3346	36.0796	43.1933	43.6417
20.43	47.5099	32.8252	57.7197	64.5247	38.8566	52.9092	63.9204
22.3596	49.5866	37.7885	58.1852	55.4753	82.0482	62.2115	48.7586
23.9621	50.8272	53.3302	50.8272	63.9204	26.5701	76.0379	40.3753
25.3452	31.3133	60.6271	31.3133	37.7885	62.2115	58.6575	52.9092
26.5701	70.379	79.5701	40.8629	45.8207	58.6575	55.0398	44.5248
27.6751	32.8252	36.6654	52.9092	31.3133	44.9602	44.5248	32.8252
28.686	55.9146	31.3133	73.4299	36.6654	37.7885	40.8629	50
29.621	52.4901	59.1371	14.3093	59.6248	32.0885	48.3435	52.4901
30.4934	58.6575	72.3249	60.6271	49.1728	17.9518	53.3302	55.0398
31.3133	51.6565	69.5066	45.3921	46.6698	76.0379	33.5284	37.7885
32.0885	58.1852	47.9274	39.8789	29.621	31.3133	57.7197	36.0796
32.8252	33.5284	48.7586	62.2115	41.8148	28.686	34.8503	59.6248
33.5284	57.2604	28.686	46.6698	30.4934	56.3583	28.686	49.5866
34.2024	38.3288	48.3435	46.2465	71.314	65.1497	43.6417	76.0379
34.8503	40.8629	52.0726	65.7977	66.4716	49.1728	85.6907	60.1211
35.4753	48.3435	85.6907	64.5247	47.5099	55.9146	34.2024	54.1793
36.0796	69.5066	36.0796	70.379	53.3302	50	47.0908	23.9621
36.6654	67.9116	62.2115	63.9204	58.1852	70.379	57.2604	61.1435
37.2345	54.1793	49.1728	20.43	67.1749	73.4299	51.2414	42.2803
37.7885	1	39.3729	25.3452	47.0908	52.9092	59.6248	48.3435
38.3288	50.4135	61.1435	72.3249	82.0482	57.2604	42.2803	34.2024
38.8566	64.5247	46.6698	36.0796	50	60.6271	14.3093	60.6271
39.3729	44.0854	40.8629	76.0379	46.2465	60.1211	38.3288	79.5701
39.8789	59.6248	61.6712	67.1749	22.3596	62.7655	46.2465	33.5284
40.3753	77.6404	65.1497	45.8207	79.5701	48.3435	50.8272	36.6654
40.8629	25.3452	14.3093	52.4901	51.6565	44.5248	69.5066	61.6712
41.3426	44.5248	64.5247	36.6654	43.1933	65.7977	30.4934	82.0482
41.8148	67.1749	25.3452	41.8148	59.1371	41.8148	37.2345	65.7977
42.2803	36.6654	55.9146	47.9274	27.6751	46.6698	63.9204	77.6404
42.7397	53.7535	74.6548	60.1211	53.7535	59.6248	82.0482	30.4934
43.1933	38.8566	54.608	22.3596	38.3288	30.4934	41.3426	69.5066
43.6417	26.5701	63.9204	67.9116	52.0726	39.8789	62.7655	39.3729
44.0854	42.7397	45.8207	65.1497	28.686	77.6404	64.5247	41.8148
44.5248	82.0482	42.2803	62.7655	77.6404	47.9274	65.7977	47.5099
44.9602	30.4934	34.2024	77.6404	65.7977	43.1933	45.8207	41.3426
45.3921	48.7586	29.621	23.9621	39.8789	66.4716	61.1435	39.8789
45.8207	14.3093	58.1852	47.5099	56.8068	64.5247	50.4135	17.9518
46.2465	61.6712	41.3426	32.8252	62.2115	42.7397	20.43	34.8503
46.6698	23.9621	62.7655	79.5701	35.4753	54.608	58.1852	56.3583
47.0908	27.6751	44.0854	30.4934	44.0854	45.3921	32.8252	29.621
47.5099	53.3302	71.314	59.1371	14.3093	46.2465	79.5701	47.9274
47.9274	34.8503	70.379	52.0726	62.7655	40.8629	29.621	67.1749
48.3435	55.0398	43.6417	33.5284	60.6271	40.3753	73.4299	74.6548
48.7586	32.0885	26.5701	49.5866	67.9116	72.3249	54.608	45.8207
49.1728	61.1435	59.6248	27.6751	65.1497	59.1371	67.9116	62.7655
49.5866	42.2803	17.9518	50	44.5248	85.6907	36.0796	37.2345
50	40.3753	53.7535	29.621	50.8272	14.3093	77.6404	46.2465
50.4135	72.3249	27.6751	38.8566	37.2345	49.5866	74.6548	57.7197
50.8272	43.6417	23.9621	35.4753	25.3452	44.0854	42.7397	72.3249
51.2414	73.4299	63.3346	44.9602	34.2024	25.3452	39.8789	52.0726
51.6565	37.7885	82.0482	48.3435	69.5066	79.5701	53.7535	57.2604
52.0726	45.3921	40.3753	59.6248	1	34.2024	50	26.5701

Table A1.5: DoE co-ordinates for 99 points (plus one nominal point) and 8 design variables normally distributed

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
52.4901	63.3346	47.0908	66.4716	51.2414	29.621	65.1497	73.4299
52.9092	66.4716	44.5248	41.3426	17.9518	34.8503	55.9146	67.9116
53.3302	34.2024	52.4901	32.0885	50.4135	51.2414	68.6868	14.3093
53.7535	99	47.5099	1	39.3729	55.0398	51.6565	44.9602
54.1793	62.2115	33.5284	57.2604	54.608	27.6751	59.1371	20.43
54.608	60.1211	76.0379	54.1793	38.8566	54.1793	61.6712	99
55.0398	65.7977	20.43	38.3288	73.4299	53.3302	49.1728	35.4753
55.4753	79.5701	52.9092	55.9146	40.8629	71.314	23.9621	49.1728
55.9146	17.9518	46.2465	17.9518	55.0398	52.4901	45.3921	66.4716
56.3583	76.0379	51.6565	54.608	76.0379	20.43	41.8148	55.9146
56.8068	54.608	1	28.686	60.1211	67.9116	52.4901	59.1371
57.2604	85.6907	57.7197	49.1728	52.4901	32.8252	71.314	38.3288
57.7197	44.9602	67.1749	37.7885	58.6575	1	36.6654	43.1933
58.1852	43.1933	77.6404	43.1933	68.6868	55.4753	66.4716	40.8629
58.6575	46.2465	41.8148	26.5701	32.0885	63.9204	32.0885	38.8566
59.1371	74.6548	32.0885	85.6907	48.7586	48.7586	44.0854	54.608
59.6248	52.0726	65.7977	68.6868	99	50.8272	37.7885	53.3302
60.1211	35.4753	35.4753	42.7397	70.379	22.3596	56.8068	55.4753
60.6271	50	50.4135	61.1435	61.1435	61.1435	99	62.2115
61.1435	36.0796	56.8068	63.3346	20.43	42.2803	27.6751	42.7397
61.6712	51.2414	99	42.2803	42.7397	50.4135	1	50.4135
62.2115	56.3583	55.4753	82.0482	47.9274	53.7535	25.3452	68.6868
62.7655	52.9092	67.9116	51.2414	42.2803	36.6654	55.4753	1
63.3346	29.621	68.6868	53.3302	33.5284	61.6712	38.8566	63.3346
63.9204	39.8789	45.3921	56.8068	54.1793	35.4753	17.9518	71.314
64.5247	65.1497	38.8566	69.5066	55.9146	69.5066	44.9602	31.3133
65.1497	55.4753	50.8272	55.4753	72.3249	74.6548	26.5701	50.8272
65.7977	71.314	54.1793	40.3753	61.6712	47.0908	35.4753	70.379
66.4716	57.7197	73.4299	34.2024	40.3753	68.6868	47.5099	58.1852
67.1749	47.0908	34.8503	47.0908	23.9621	63.3346	72.3249	56.8068
67.9116	60.6271	58.6575	99	26.5701	52.0726	49.5866	51.2414
68.6868	68.6868	57.2604	37.2345	56.3583	58.1852	70.379	44.0854
69.5066	39.3729	43.1933	44.5248	85.6907	45.8207	48.7586	32.0885
70.379	46.6698	30.4934	61.6712	48.3435	23.9621	31.3133	47.0908
71.314	22.3596	39.8789	71.314	52.9092	57.7197	46.6698	45.3921
72.3249	63.9204	42.7397	48.7586	34.8503	56.8068	60.1211	22.3596
73.4299	41.8148	60.1211	34.8503	41.3426	67.1749	47.9274	28.686
74.6548	49.1728	50	58.6575	57.2604	47.5099	22.3596	27.6751
76.0379	37.2345	55.0398	74.6548	57.7197	38.3288	63.3346	51.6565
77.6404	56.8068	37.2345	39.3729	36.0796	33.5284	40.3753	46.6698
79.5701	41.3426	44.9602	53.7535	45.3921	99	52.0726	64.5247
82.0482	47.9274	51.2414	44.0854	74.6548	51.6565	54.1793	85.6907
85.6907	45.8207	22.3596	55.0398	43.6417	41.3426	56.3583	65.1497
99	28.686	66.4716	50.4135	44.9602	43.6417	67.1749	53.7535

Table A1.5: ***(continued)***

A1.2 PHYSICAL PROGRAMMING METHOD EXAMPLE

Example transformation for the 'smaller-the-better' scenario

A designer specifies that a desirable range for the mean performance is between 3 and 5, i.e. the lower bound (LB) for the desirable range is $g_{i1} = 3$ and the upper bound (UB) is $g_{i2} = 5$. The current objective function value is $f(i) = 3.8$ so the solution is in the desirable range. To map this onto the class function firstly requires mapping the range, $g_{ip} < f(i) \le g_{iq}$, to the transformed range, $p < g_i \le q$ where $f(i)$ transforms to g_i as can be seen below:

$$\begin{aligned} &\text{Range:} && g_{ip} < f(i) \le g_{iq} \\ &\text{Transformed range:} && p < g_i \le q \end{aligned} \tag{A1.1}$$

In this example (for the desirable range) the LB, $g_{i1} = 3$, so $p = 1$ and the UB, $g_{i2} = 5$, so $q = 2$. This means a transformation of the desirable range from 3-to-5 to 1-to-2. $f(i)$ is then mapped to a point g_i in the interval range 1 to 2. In Figure 5 it can be seen that $f(i) = 3.8$ transforms to $g_i = 1.4$ and the class function, $\overline{g}_i$ then becomes $1.4^2 = 1.96$ using Equation (3.5.6). The method uses the following steps (shown graphically in Figure A1.6):

1. Get current objective function, here $f(i) = 3.8$
2. Determine in which preference range $f(i)$ is located. In this example $g_{i1} < f(i) \le g_{i2}$ so $f(i)$ is located in the desirable range.
3. Determine the location, L of $f(i)$ in the chosen range using the following equation

$$L = \frac{f(i) - g_{ip}}{g_{iq} - g_{ip}} \tag{A1.2}$$

For the example L becomes:

$$L = \frac{3.8 - 3}{5 - 3} = 0.4$$

4. Locate L in the transformed range ($p < g_i \le q$) so that g_i is calculated as follows

$$g_i = L + p\,. \tag{A1.3}$$

In this example, where the desirable range is $g_{i1} < f(i) \leq g_{i2}$ and by Equation (A1.1) $p = 1$ and $q = 2$, $g_i = 1.4$.

5. Calculate the value of the class function $\overline{g}_i$ using Equation (3.5.6) so that

$$\overline{g}_i = 1.4^2 = 1.96$$

6. Repeat steps 1-5 for each of the *n* objectives.
7. Calculate the aggregate objective function (AOF), F used by the optimizer by the following equation

$$F = \sum_{i=1}^{n} \overline{g}_i \qquad \text{(A1.4)}$$

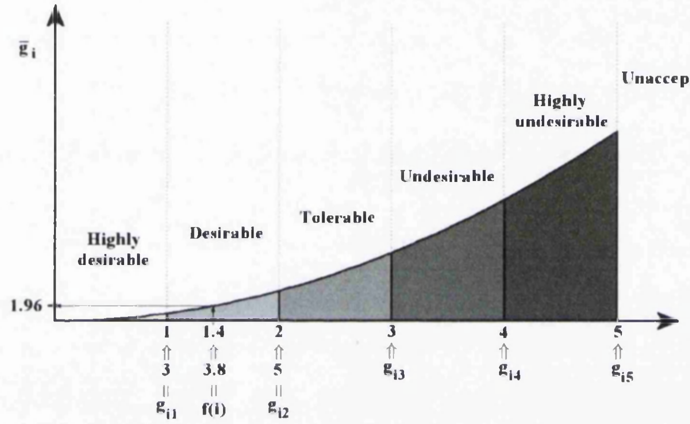


Figure A1.6: Example transformation of $f(i)$ to $\overline{g}_i$

APPENDIX A2

ROBUST DESIGN OPTIMIZATION PROCEDURE

This appendix gives the structure and the source files of the RDO procedure used in Chapter 5 and discussed in Chapter 4.

A2.1 STRUCTURE

This appendix gives a description of the robust design procedure implemented in this work.

The overall structure can be seen in Figure A2.1. It can be seen, that the script 'optimize.pl' starts the procedure, then the initialization procedure is called. The initialization procedure is shown in Figure A2.2. After initialization, the optimization starts and, depending on the number of points n required on the Pareto front, n optimizations are run by calling OPTIMIZER.exe, this is the optimizer and can be any optimizer, in this example it is DOT. The structure of OPTIMIZER.exe is shown in Figure A2.3.

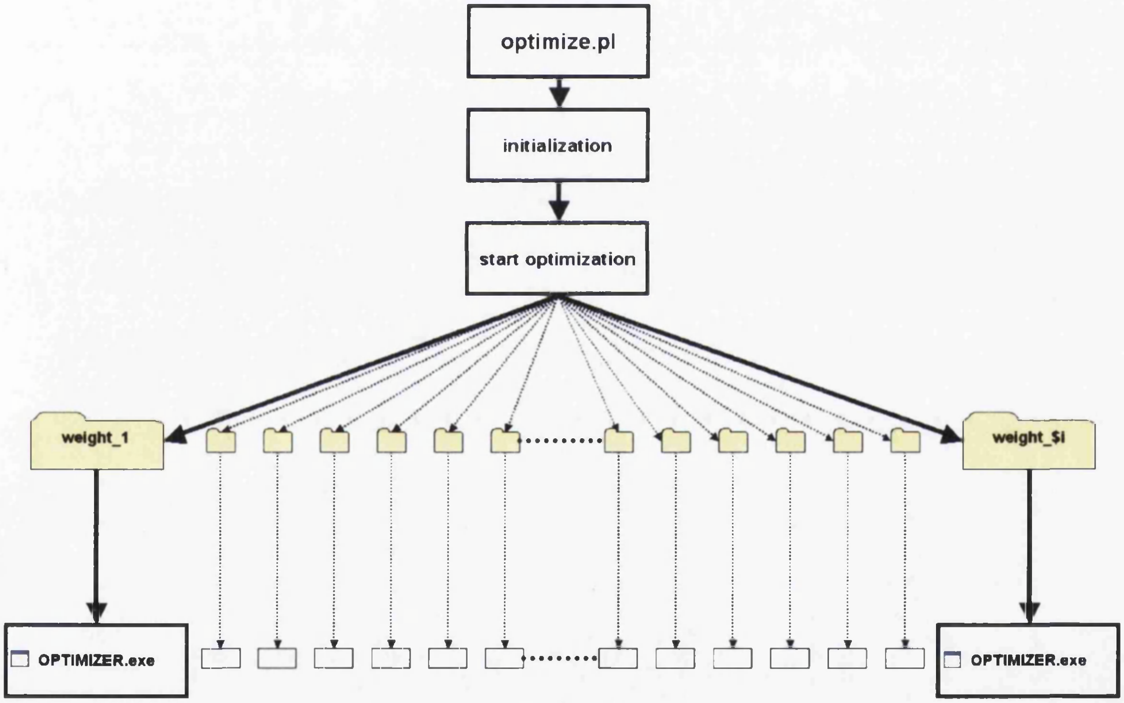


Figure A2.1: Overall structure of the RDO process

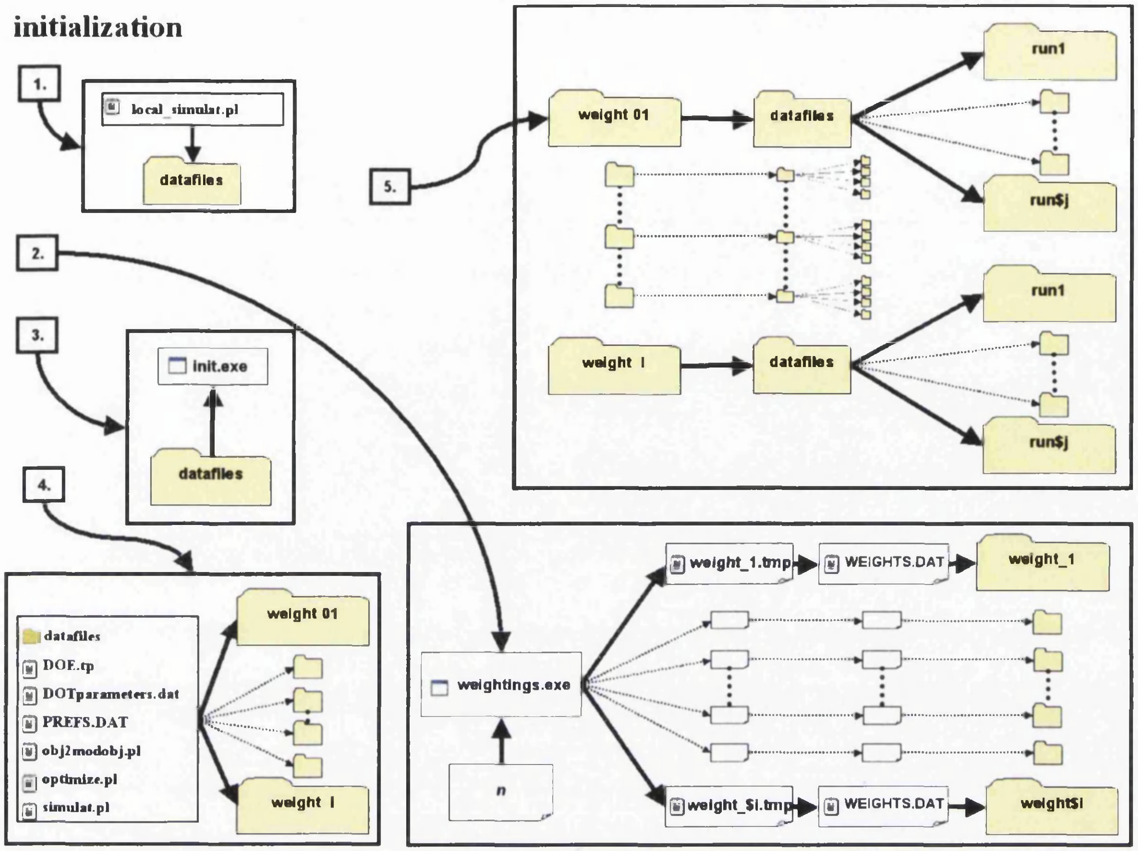


Figure A2.2: initialization procedure

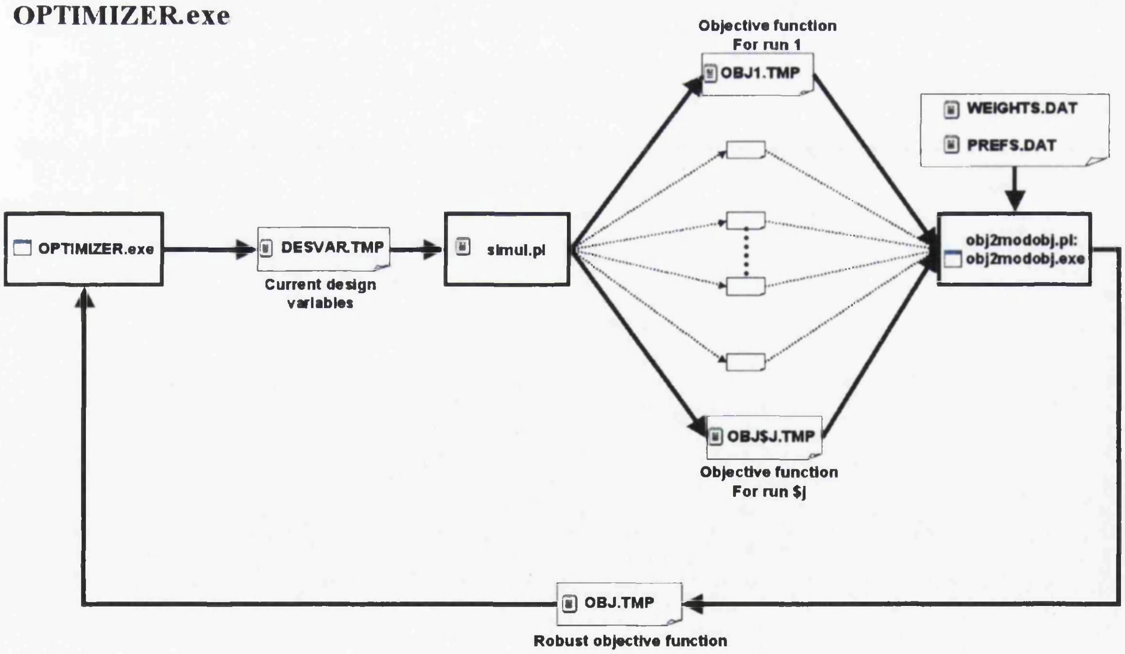


Figure A2.3: OPTIMIZER.exe

Once OPTIMIZER.exe is invoked the current design variables are generated and the script simul.pl is called. The structure of simul.pl is shown in Figure A2.4, the source code is listed in section A2.2.

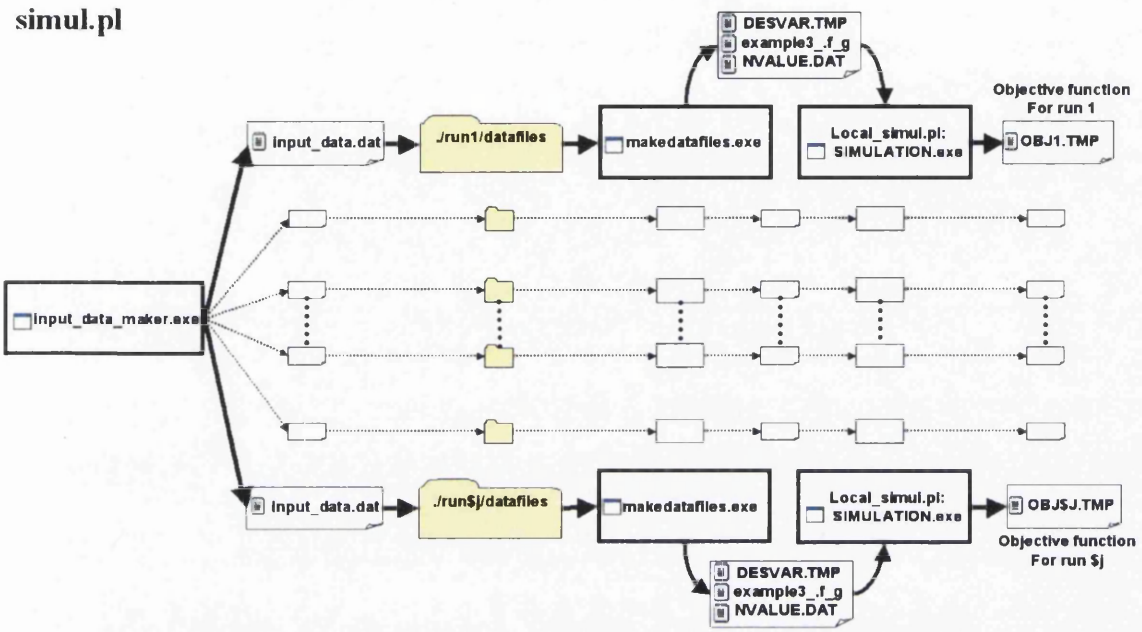
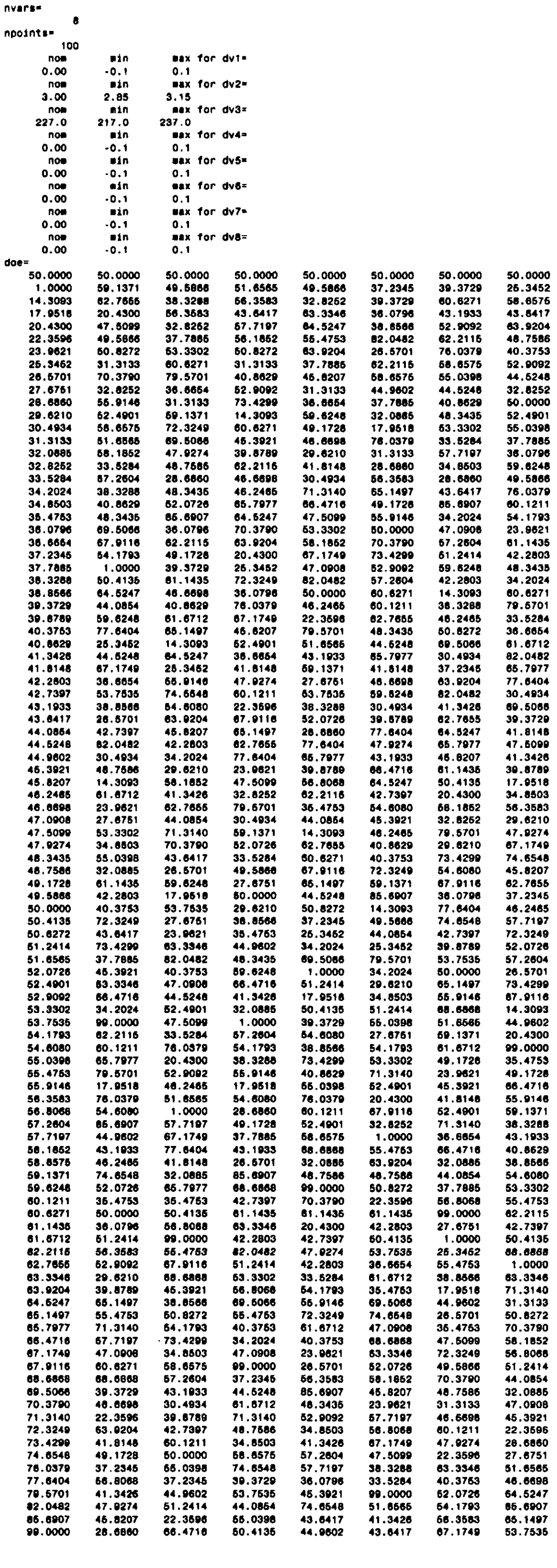


Figure A2.4: simul.pl

A2.2 SOURCE CODE AND EXAMPLE DATA FILE

This section contains the source code and data files for the execution of the RDO structure described in Section A2.1.



```
nvars=
        8
npoints=
      100
      nom        min        max for dv1=
     0.00       -0.1        0.1
      nom        min        max for dv2=
     3.00       2.85       3.15
      nom        min        max for dv3=
    227.0      217.0      237.0
      nom        min        max for dv4=
     0.00       -0.1        0.1
      nom        min        max for dv5=
     0.00       -0.1        0.1
      nom        min        max for dv6=
     0.00       -0.1        0.1
      nom        min        max for dv7=
     0.00       -0.1        0.1
      nom        min        max for dv8=
     0.00       -0.1        0.1
doe=
   50.0000    50.0000    50.0000    50.0000    50.0000    50.0000    50.0000    50.0000
    1.0000    59.1371    49.5866    51.6565    49.5866    37.2345    39.3729    25.3452
   14.3093    62.7655    38.3288    56.3583    32.8252    39.3729    60.6271    58.6575
   17.9518    20.4300    56.3583    43.6417    63.3346    36.0796    43.1933    43.6417
   20.4300    47.5099    32.8252    57.7197    64.5247    38.8566    52.9092    63.9204
   22.3596    49.5866    37.7885    58.1852    55.4753    82.0482    62.2115    48.7586
   23.9621    50.8272    53.3302    50.8272    63.9204    26.5701    76.0379    40.3753
   25.3452    31.3133    60.6271    31.3133    37.7885    62.2115    58.6575    52.9092
   26.5701    70.3790    79.5701    40.8629    45.8207    58.6575    55.0398    44.5248
   27.6751    32.8252    36.6654    52.9092    31.3133    44.9602    44.5248    32.8252
   28.6860    55.9146    31.3133    73.4299    36.6654    37.7885    40.8629    50.0000
   29.6210    52.4901    59.1371    14.3093    59.6248    32.0885    48.3435    52.4901
   30.4934    58.6575    72.3249    60.6271    49.1728    17.9518    53.3302    55.0398
   31.3133    51.6565    69.5066    45.3921    46.6698    76.0379    33.5284    37.7885
   32.0885    58.1852    47.9274    39.8789    29.6210    31.3133    57.7197    36.0796
   32.8252    33.5284    48.7586    62.2115    41.8148    28.6860    34.8503    59.6248
   33.5284    57.2604    28.6860    46.6698    30.4934    56.3583    28.6860    49.5866
   34.2024    38.3288    48.3435    46.2465    71.3140    65.1497    43.6417    76.0379
   34.8503    40.8629    52.0726    65.7977    66.4716    49.1728    85.6907    60.1211
   35.4753    48.3435    85.6907    64.5247    47.5099    55.9146    34.2024    54.1793
   36.0796    69.5066    36.0796    70.3790    53.3302    50.0000    47.0908    23.9621
   36.6654    67.9116    62.2115    63.9204    58.1852    70.3790    57.2604    61.1435
   37.2345    54.1793    49.1728    20.4300    67.1749    73.4299    51.2414    42.2803
   37.7885     1.0000    39.3729    25.3452    47.0908    52.9092    59.6248    48.3435
   38.3288    50.4135    61.1435    72.3249    82.0482    57.2604    42.2803    34.2024
   38.8566    64.5247    46.6698    36.0796    50.0000    60.6271    14.3093    60.6271
   39.3729    44.0854    40.8629    76.0379    46.2465    60.1211    38.3288    79.5701
   39.8789    59.6248    61.6712    67.1749    22.3596    62.7655    46.2465    33.5284
   40.3753    77.6404    65.1497    45.8207    79.5701    48.3435    50.8272    36.6654
   40.8629    25.3452    14.3093    52.4901    51.6565    44.5248    69.5066    61.6712
   41.3426    44.5248    64.5247    36.6654    43.1933    65.7977    30.4934    82.0482
   41.8148    67.1749    25.3452    41.8148    59.1371    41.8148    37.2345    65.7977
   42.2803    36.6654    55.9146    47.9274    27.6751    46.6698    63.9204    77.6404
   42.7397    53.7535    74.6548    60.1211    53.7535    59.6248    82.0482    30.4934
   43.1933    38.8566    54.6080    22.3596    38.3288    30.4934    41.3426    69.5066
   43.6417    26.5701    63.9204    67.9116    52.0726    39.8789    62.7655    39.3729
   44.0854    42.7397    45.8207    65.1497    28.6860    77.6404    64.5247    41.8148
   44.5248    82.0482    42.2803    62.7655    77.6404    47.9274    65.7977    47.5099
   44.9602    30.4934    34.2024    77.6404    65.7977    43.1933    45.8207    41.3426
   45.3921    48.7586    29.6210    23.9621    39.8789    66.4716    61.1435    39.8789
   45.8207    14.3093    58.1852    47.5099    56.8068    64.5247    50.4135    17.9518
   46.2465    61.6712    41.3426    32.8252    62.2115    42.7397    20.4300    34.8503
   46.6698    23.9621    62.7655    79.5701    35.4753    54.6080    58.1852    56.3583
   47.0908    27.6751    44.0854    30.4934    44.0854    45.3921    32.8252    29.6210
   47.5099    53.3302    71.3140    59.1371    14.3093    46.2465    79.5701    47.9274
   47.9274    34.8503    70.3790    52.0726    62.7655    40.8629    29.6210    67.1749
   48.3435    55.0398    43.6417    33.5284    60.6271    40.3753    73.4299    74.6548
   48.7586    32.0885    26.5701    49.5866    67.9116    72.3249    54.6080    45.8207
   49.1728    61.1435    59.6248    27.6751    65.1497    59.1371    67.9116    62.7655
   49.5866    42.2803    17.9518    50.0000    44.5248    85.6907    36.0796    37.2345
   50.0000    40.3753    53.7535    29.6210    50.8272    14.3093    77.6404    46.2465
   50.4135    72.3249    27.6751    38.8566    37.2345    49.5866    74.6548    57.7197
   50.8272    43.6417    23.9621    35.4753    25.3452    44.0854    42.7397    72.3249
   51.2414    73.4299    63.3346    44.9602    34.2024    25.3452    39.8789    52.0726
   51.6565    37.7885    82.0482    48.3435    69.5066    79.5701    53.7535    57.2604
   52.0726    45.3921    40.3753    59.6248     1.0000    34.2024    50.0000    26.5701
   52.4901    63.3346    47.0908    66.4716    51.2414    29.6210    65.1497    73.4299
   52.9092    66.4716    44.5248    41.3426    17.9518    34.8503    55.9146    67.9116
   53.3302    34.2024    52.4901    32.0885    50.4135    51.2414    68.6868    14.3093
   53.7535    99.0000    47.5099     1.0000    39.3729    55.0398    51.6565    44.9602
   54.1793    62.2115    33.5284    57.2604    54.6080    27.6751    59.1371    20.4300
   54.6080    60.1211    76.0379    54.1793    38.8566    54.1793    61.6712    99.0000
   55.0398    65.7977    20.4300    38.3288    73.4299    53.3302    49.1728    35.4753
   55.4753    79.5701    52.9092    55.9146    40.8629    71.3140    23.9621    49.1728
   55.9146    17.9518    46.2465    17.9518    55.0398    52.4901    45.3921    66.4716
   56.3583    76.0379    51.6565    54.6080    76.0379    20.4300    41.8148    55.9146
   56.8068    54.6080     1.0000    28.6860    60.1211    67.9116    52.4901    59.1371
   57.2604    85.6907    57.7197    49.1728    52.4901    32.8252    71.3140    38.3288
   57.7197    44.9602    67.1749    37.7885    58.6575     1.0000    36.6654    43.1933
   58.1852    43.1933    77.6404    43.1933    68.6868    55.4753    66.4716    40.8629
   58.6575    46.2465    41.8148    26.5701    32.0885    63.9204    32.0885    38.8566
   59.1371    74.6548    32.0885    85.6907    48.7586    48.7586    44.0854    54.6080
   59.6248    52.0726    65.7977    68.6868    99.0000    50.8272    37.7885    53.3302
   60.1211    35.4753    35.4753    42.7397    70.3790    22.3596    56.8068    55.4753
   60.6271    50.0000    50.4135    61.1435    61.1435    61.1435    99.0000    62.2115
   61.1435    36.0796    56.8068    63.3346    20.4300    42.2803    27.6751    42.7397
   61.6712    51.2414    99.0000    42.2803    42.7397    50.4135     1.0000    50.4135
   62.2115    56.3583    55.4753    82.0482    47.9274    53.7535    25.3452    68.6868
   62.7655    52.9092    67.9116    51.2414    42.2803    36.6654    55.4753     1.0000
   63.3346    29.6210    68.6868    53.3302    33.5284    61.6712    38.8566    63.3346
   63.9204    39.8789    45.3921    56.8068    54.1793    35.4753    17.9518    71.3140
   64.5247    65.1497    38.8566    69.5066    55.9146    69.5066    44.9602    31.3133
   65.1497    55.4753    50.8272    55.4753    72.3249    74.6548    26.5701    50.8272
   65.7977    71.3140    54.1793    40.3753    61.6712    47.0908    35.4753    70.3790
   66.4716    57.7197    73.4299    34.2024    40.3753    68.6868    47.5099    58.1852
   67.1749    47.0908    34.8503    47.0908    23.9621    63.3346    72.3249    56.8068
   67.9116    60.6271    58.6575    99.0000    26.5701    52.0726    49.5866    51.2414
   68.6868    68.6868    57.2604    37.2345    56.3583    58.1852    70.3790    44.0854
   69.5066    39.3729    43.1933    44.5248    85.6907    45.8207    48.7586    32.0885
   70.3790    46.6698    30.4934    61.6712    48.3435    23.9621    31.3133    47.0908
   71.3140    22.3596    39.8789    71.3140    52.9092    57.7197    46.6698    45.3921
   72.3249    63.9204    42.7397    48.7586    34.8503    56.8068    60.1211    22.3596
   73.4299    41.8148    60.1211    34.8503    41.3426    67.1749    47.9274    28.6860
   74.6548    49.1728    50.0000    58.6575    57.2604    47.5099    22.3596    27.6751
   76.0379    37.2345    55.0398    74.6548    57.7197    38.3288    63.3346    51.6565
   77.6404    56.8068    37.2345    39.3729    36.0796    33.5284    40.3753    46.6698
   79.5701    41.3426    44.9602    53.7535    45.3921    99.0000    52.0726    64.5247
   82.0482    47.9274    51.2414    44.0854    74.6548    51.6565    54.1793    85.6907
   85.6907    45.8207    22.3596    55.0398    43.6417    41.3426    56.3583    65.1497
   99.0000    28.6860    66.4716    50.4135    44.9602    43.6417    67.1749    53.7535
```

DoE.tp - for 8 design variables and 100 points

```
#!/usr/bin/perl

# remove previous overall results folder
system ("rm -r -f results");

# make new overall results folder
system ("mkdir results");

# move to utopia_f1 directory and extract results
chdir ("./utopia_f1/results");

#make INFO.DAT file

my $filename = "INFO.DAT";
open NEWFILE, ">$filename";
print NEWFILE "f1   100";
close NEWFILE;

#collate results
system("../../source/get_results.exe");

#remove old results file
system("rm -f -r results_f1.res");

# copy the results.res file to result file corresponding to weighting
system("cp -r results.res results_f1.res");

# move the results to the overall results folder
system("mv  results_f1.res ../../results");

# remove results.res file to save disk space
system("rm -f results.res");

# move to utopia_f2 directory and extract results
chdir ("../../utopia_f2/results");

#make INFO.DAT file

my $filename = "INFO.DAT";
open NEWFILE, ">$filename";
print NEWFILE "f2   100";
close NEWFILE;

#collate results
system("../../source/get_results.exe");

#remove old results file
system("rm -f -r results_f2.res");

# copy the results.res file to result file corresponding to weighting
system("cp -r results.res results_f2.res");

# move the results to the overall results folder
system("mv  results_f2.res ../../results");

# remove results.res file to save disk space
system("rm -f results.res");

chdir("../../");
die;
#now do the same for each weighting

for ($i=1;$j<31;$j++)
{
        # move to weight $i directory and extract results
        chdir ("./weight$i/results");

        #make INFO.DAT file

        my $filename = "INFO.DAT";
        open NEWFILE, ">$filename";
        print NEWFILE "$i   100";
        close NEWFILE;

        #collate results
        system("../../source/get_results.exe");

        #remove old results file
        system("rm -f -r results_weight$i\.res");

        # copy the results.res file to result file corresponding to weighting
        system("cp -r results.res results_weight$i\.res");

        # move the results to the overall results folder
        system("mv  results_weight$i\.res ../../results");

        # remove results.res file to save disk space
        system("rm -f results.res");

        chdir("../../");
}
```

get_res_bat.pl

```
#!/usr/bin/perl

$i=$ARGV[0];
print("$i");
system("../../../../../Scott/slit/slitsim_nin");
system("mv OBJ.TMP OBJ$i\.TMP");
system("mv  OBJ$i\.TMP ../../");
```

local_simulat.pl

```
#!/usr/bin/perl

system("sleep 1");
system("nice -20 ../source/wait.exe");
system("../source/obj2modobj.exe");
```

obj2modobj.pl

```
#!/usr/bin/perl

#copy local_simlat.pl to datafiles
system("cp -r local_simulat.pl datafiles");

#remove all weights directories
system("rm -r -f weight_*");

for ($i=30;$i>0;$i--)
{
  # make weights and results directories
  system("mkdir weight_$i");
  system("mkdir weight_$i/results");
}
  # Calculate the weights
  system("./source/weightings.exe");
 # Now there n files named weights_$i.tmp

for ($i=30;$i>0;$i--)
{
  # change name of  weight_$i.tmp to WEIGHTS.DAT
  system("mv WEIGHTS_$i\.TMP WEIGHTS.DAT");
  system("mv WEIGHTS.DAT weight_$i");
}

# loop through all the weights starting from 30 and run optimization for each weight
for ($i=30;$i>0;$i--)
{
  # in parent directory
  # move to datafiles directory
    chdir ("datafiles");
    # in directory datafiles
    #initialize the results files by calling init
               system("../../../Scott/init/init");
    chdir("../");
    # out of directory datafiles

  # copy 'datafiles' folder, simulat.pl, obj2modobj.pl, DOE.tp, DOTparameters.dat and PREFS.DAT to folder weight_$i
  system("cp -r datafiles simulat.pl obj2modobj.pl DOE.tp DOTparameters.dat PREFS.DAT weight_$i");

  # move to directory weight_$i
  chdir("weight_$i");
  # in directory weight_$i, start with initialization

        #initialization start

        # make a copy of example3.f_g to example3.tmp so that this can be modified in makedatafiles.exe
        # to create the current example3.f_g file
        system("cp ./datafiles/example3.f_g ./datafiles/example3.tmp");
        system("rm -f ./datafiles/opt.res");

        # if at the start then copy STARTX.DAT to DESVAR.TMP to give a start point for the optimization
        if($i==30)
        {
                system("cp ../STARTX.DAT DESVAR.TMP");
        }

        # for all other weights use previous solution by executing previousbest.exe to generate DESVAR.TMP which becomes
        # the start point of optimization
        if($i<30)
        {
                system("../source/previousbest.exe");
        }
```

optimize.pl

```
        # remove any old run directories and make new directories
        for ($j=1;$j<101;$j++)
        {
                system("rm -r -f run$j");
                system("mkdir run$j");
        }

        # loop from 1 to the number of runs (100)
        for ($j=1;$j<101;$j++)
        {
                # copy the current datafiles directory to the run$j directory
                system("cp -r datafiles run$j");
        }
        #initialization finished
        # run DOT
        system("pwd");
        system("../DOT/DOToptimize.EXEC");
        # DOT finished
        # Now for PFM-PP need to use the solution just found and make this the start point for the next weight
        system("cp results_usingGA_a.res previous_results.res");
        $k=$i-1;
        system("mv previous_results.res ../weight_$k");

        # move results_usingGA_a.res and results_usingGA_b.res to the results folder
        system("mv results_usingGA_a.res results");
        system("mv results_usingGA_b.res results");

        # move results and delete datafiles folders for each run
        for ($j=1;$j<101;$j++)
        {
                chdir("run$j");
                        chdir("datafiles");
                                system("cp HSFDieLipXFlowPerStep.res HSFDieLipXFlowPerStep_run$j\.res");
                                system("mv HSFDieLipXFlowPerStep_run$j\.res  ../../results");

                                # run 1 is nominal run and so copy chokber bar results file to results only for the nominal
                                if($j==1)
                                {
                                        system("mv ChkBarNodesPerStep.res ../../results");
                                }
                                system("mv example3.f_g ../");
                                system("mv NVALUE.DAT ../");
                                system("mv DESVAR.TMP ../");
                                system("mv opt.res ../");
                        chdir("../");
                        system("rm -r -f datafiles");
                chdir("../");
        }

        # remove datafiles folder from weight directory
        system("rm -r -f datafiles");
  chdir("../");
}
```

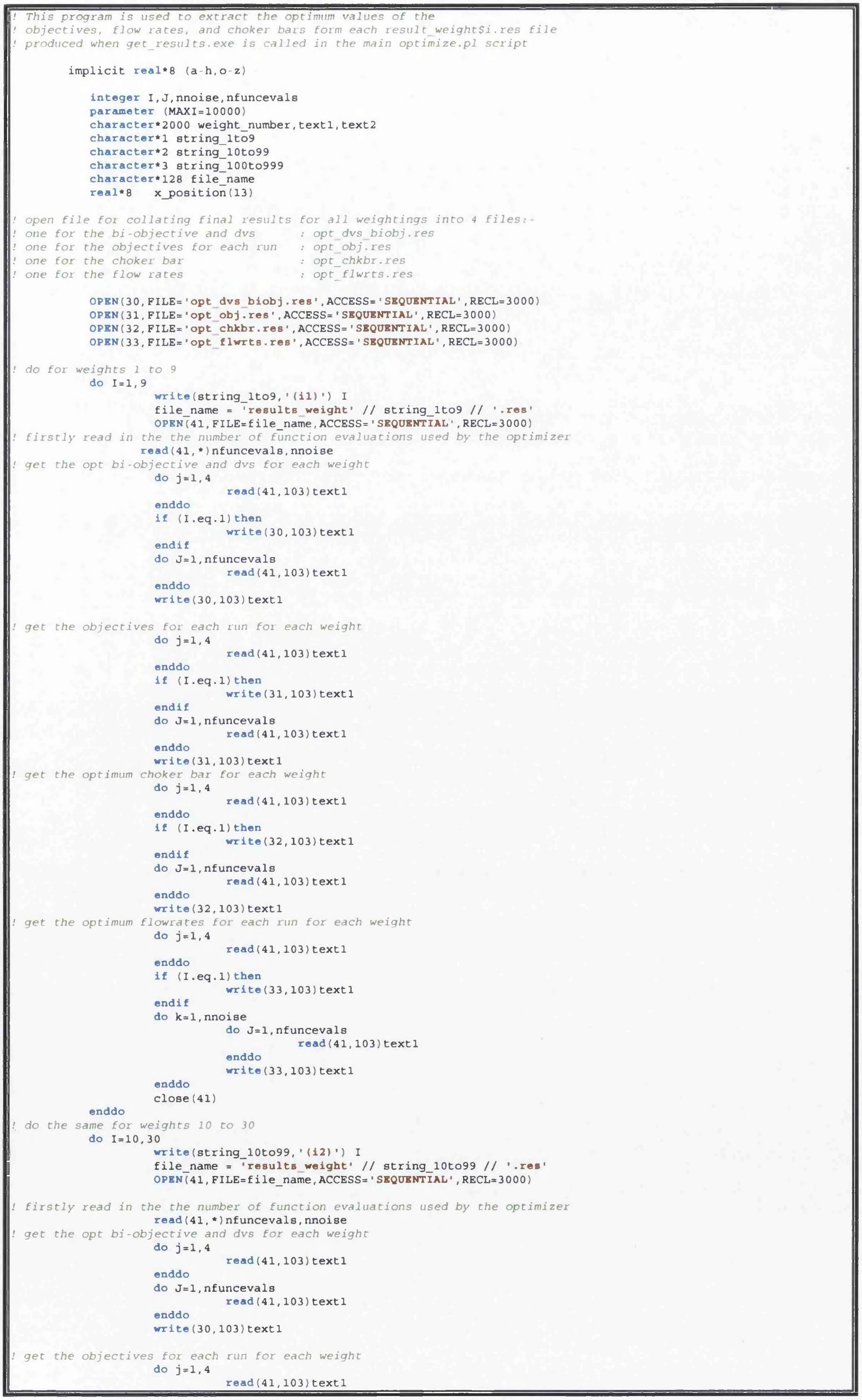
optimize.pl *(continued)*

```
#!/usr/bin/perl

# Now create input_data$i.dat files for each run

system("cp -r ../counter.ini counter.tmp");

for ($i=1;$i<101;$i++)
{
        system("../source/input_data_maker.exe");
        system("cp input_data.dat input_data$i\.dat ");
}

# modify datafiles for the current simulations
for ($i=1;$i<101;$i++)
{
        # copy input_data$i.dat to input_data.dat and move this into the run$ folder
        system("cp input_data$i\.dat input_data.dat");
        system("rm -f input_data$i\.dat");
        system("mv input_data.dat run$i");

        # move to run$i directory and copy input_data.dat to datafiles
        chdir("run$i");
                system("cp input_data.dat datafiles");
                chdir("datafiles");
                        # call makedatafiles.exe which modifies DESVAR.TMP, NVALUE.DAT and example3.f_g
                        system("../../../source/makedatafiles.exe");
        chdir("../../");
        # out of run$i/datafiles and in weight directory
}

# do simulations for each run
# There are a total of 100 runs. To save time run these in parallel
# realistically only 10 at a time so break runs up into groups of 10
# with the first nine done in background and the last done in the foreground
# otherwise will cause all 100 runs to be done in background
#

# firstly remove old obj*.TMP files
system("rm -r -f OB*.TMP");


# do runs 1-9 in background then do run 10 in foreground
for ($i=1;$i<10;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=10;$i<11;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=11;$i<20;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=20;$i<21;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=21;$i<30;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=30;$i<31;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=31;$i<40;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=40;$i<41;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=41;$i<50;$i++)
{
```

simulat.pl

```
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=50;$i<51;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=51;$i<60;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=60;$i<61;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=61;$i<70;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=70;$i<71;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=71;$i<80;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=80;$i<81;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=81;$i<90;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
for ($i=90;$i<91;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i ");
        chdir ("../../");
}
for ($i=91;$i<101;$i++)
{
        chdir("run$i");
                system("pwd");
                chdir("datafiles");
                        system("local_simulat.pl $i &");
        chdir ("../../");
}
```

simulat.pl ***(continued)***



```
! This program is used to extract the optimum values of the
! objectives, flow rates, and choker bars form each result_weight$i.res file
! produced when get_results.exe is called in the main optimize.pl script

        implicit real*8 (a-h,o-z)

           integer I,J,nnoise,nfuncevals
           parameter (MAXI=10000)
           character*2000 weight_number,text1,text2
           character*1 string_1to9
           character*2 string_10to99
           character*3 string_100to999
           character*128 file_name
           real*8   x_position(13)

! open file for collating final results for all weightings into 4 files:-
! one for the bi-objective and dvs      : opt_dvs_biobj.res
! one for the objectives for each run   : opt_obj.res
! one for the choker bar                : opt_chkbr.res
! one for the flow rates                : opt_flwrts.res

           OPEN(30,FILE='opt_dvs_biobj.res',ACCESS='SEQUENTIAL',RECL=3000)
           OPEN(31,FILE='opt_obj.res',ACCESS='SEQUENTIAL',RECL=3000)
           OPEN(32,FILE='opt_chkbr.res',ACCESS='SEQUENTIAL',RECL=3000)
           OPEN(33,FILE='opt_flwrts.res',ACCESS='SEQUENTIAL',RECL=3000)

! do for weights 1 to 9
           do I=1,9
                    write(string_1to9,'(i1)') I
                    file_name = 'results_weight' // string_1to9 // '.res'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL',RECL=3000)
! firstly read in the the number of function evaluations used by the optimizer
                  read(41,*)nfuncevals,nnoise
! get the opt bi-objective and dvs for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    if (I.eq.1)then
                              write(30,103)text1
                    endif
                    do J=1,nfuncevals
                              read(41,103)text1
                    enddo
                    write(30,103)text1

! get the objectives for each run for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    if (I.eq.1)then
                              write(31,103)text1
                    endif
                    do J=1,nfuncevals
                              read(41,103)text1
                    enddo
                    write(31,103)text1
! get the optimum choker bar for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    if (I.eq.1)then
                              write(32,103)text1
                    endif
                    do J=1,nfuncevals
                              read(41,103)text1
                    enddo
                    write(32,103)text1
! get the optimum flowrates for each run for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    if (I.eq.1)then
                              write(33,103)text1
                    endif
                    do k=1,nnoise
                              do J=1,nfuncevals
                                        read(41,103)text1
                              enddo
                              write(33,103)text1
                    enddo
                    close(41)
           enddo
! do the same for weights 10 to 30
           do I=10,30
                    write(string_10to99,'(i2)') I
                    file_name = 'results_weight' // string_10to99 // '.res'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL',RECL=3000)

! firstly read in the the number of function evaluations used by the optimizer
                    read(41,*)nfuncevals,nnoise
! get the opt bi-objective and dvs for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    do J=1,nfuncevals
                              read(41,103)text1
                    enddo
                    write(30,103)text1

! get the objectives for each run for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    do J=1,nfuncevals
                              read(41,103)text1
                    enddo
                    write(31,103)text1

! get the optimum choker bar  for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    do J=1,nfuncevals
                              read(41,103)text1
                    enddo
                    write(32,103)text1
! get the optimum flowrates for each run for each weight
                    do j=1,4
                              read(41,103)text1
                    enddo
                    do k=1,nnoise
                              do J=1,nfuncevals
                                        read(41,103)text1
                              enddo
                              write(33,103)text1
                    enddo
                    close(41)
           enddo
        close(30)
          close(31)
          close(32)
          close(33)
103 format (A2000)
end
```

get_opt_results.f90

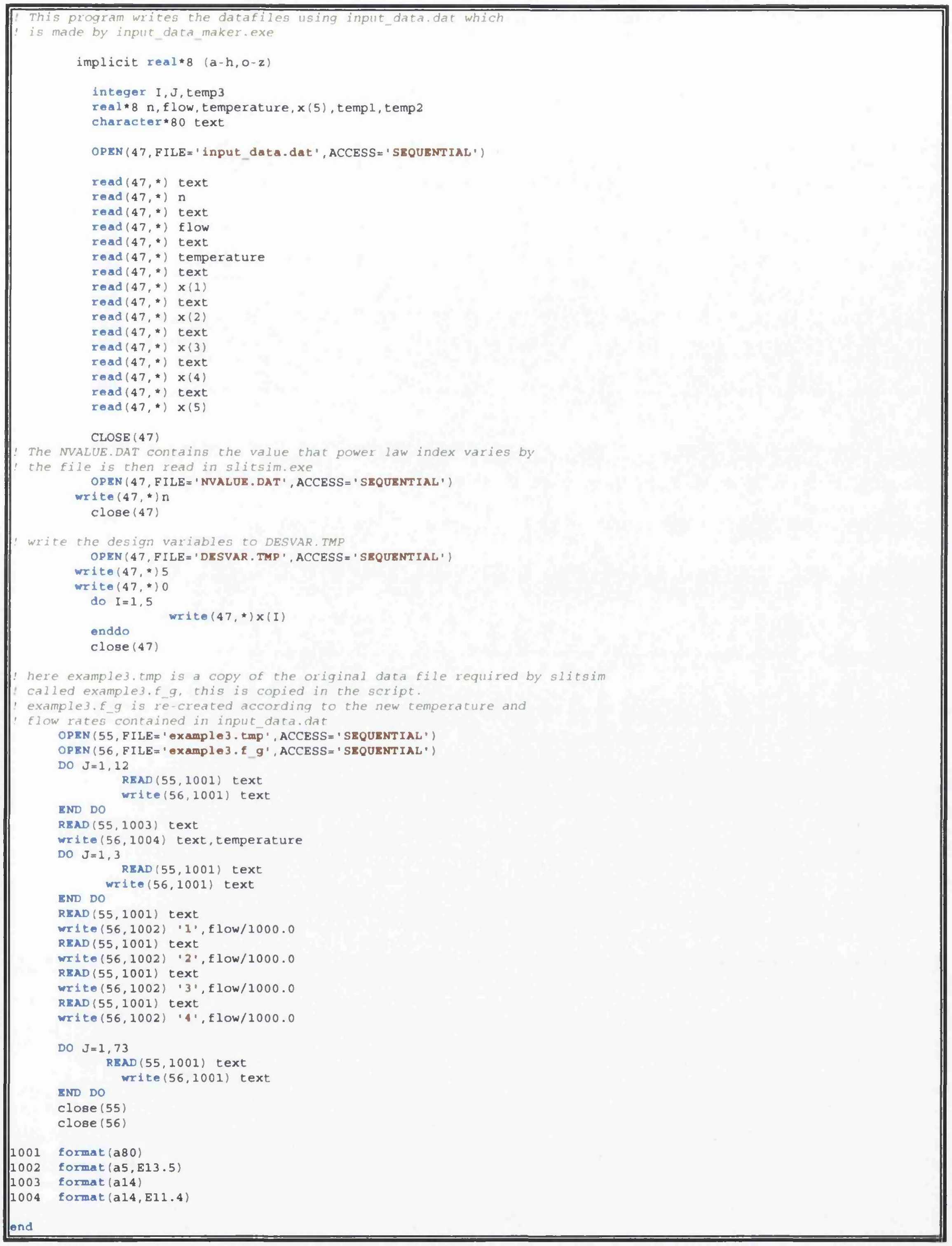
```
! This program collates all of the results from each run in each
! weighting into a file called results.res. results.res is then re-named
! outside of the program in the script to results_weight$i.res according
! to the weight.

         implicit real*8 (a-h,o-z)

           parameter (MAXI=10000)
           character*2000 weight_number,text1,text2
           character*1 string_1to9
           character*2 string_10to99
           character*3 string_100to999
           character*128 file_name
           real*8   x_position(13)
         integer I,J,nnoise,nfuncevals

! open file which contains the weight number
           OPEN(47,FILE='INFO.DAT',ACCESS='SEQUENTIAL',RECL=3000)
                   read(47,*)weight_number,nnoise
           close(47)

! open file for collating all results into one for current weighting

           OPEN(47,FILE='results.res',ACCESS='SEQUENTIAL',RECL=3000)

! calculate the number of function evaluations
           OPEN(48,FILE='results_usingGA_a.res',ACCESS='SEQUENTIAL')
           do I=1,MAXI
                   read(48,101,END=499) text1
           enddo
499        close(48)
           nfuncevals=I-2
           write(47,*) nfuncevals, nnoise

! read the objective function results, design variable values

          OPEN(48,FILE='results_usingGA_a.res',ACCESS='SEQUENTIAL')
          write(47,*)'**********************************************'
          write(47,*)'       DVs_and_OVERALL_OBJECTIVE_VALUES        '
          write(47,*)'**********************************************'
        do I=1,MAXI
                  read(48,101,END=500) text1
                  if(i.eq.1)then
                          write(47,106)'dvsobj * * * * 0',text1
                  else
                            write(47,104)'dvsobj ','weight ',weight_number,'---',0,' fcn_call# ',I-1,text1
                  endif
           enddo
500        close(48)

          OPEN(48,FILE='results_usingGA_b.res',ACCESS='SEQUENTIAL')
          write(47,*)'**********************************************'
          write(47,*)'     OBJECTIVE_VALUES_FOR_EACH_POINT_OF_DOE   '
          write(47,*)'**********************************************'
          do I=1,MAXI
                  read(48,101,END=505) text1
                  if(i.eq.1)then
                            write(47,106)runobj * * * * 0',text1
                  else
                            write(47,104)'runobj ','weight ',weight_number,'----',0,' fcn_call# ',I-1,text1
                  endif
          enddo
505       close(48)
```

```
! for choker bar results the first row contains the distance from the choker bar
! centre called x_position(i)
          do i=1,13
                    if(i.eq.1)then
                              x_position(i)=0.0e-10
                    else
                              x_position(i)=x_position(i-1)+0.05
                    endif
          enddo

! read the choker bar profile for each function evaluation for run number 1 (the nominal run)
          write(47,*)'**********************************************'
          write(47,*)'                  CHOKERBAR                   '
          write(47,*)'**********************************************'
          write(47,102)'chkbar * * * * 0',(x_position(I),I=1,13)
        OPEN(41,FILE='ChkBarNodesPerStep.res',ACCESS='SEQUENTIAL')
          do I=1,MAXI
                    read(41,103,END=501) text1
                    write(47,104)'chkbar ','weight ',weight_number,'run#',1,' fcn_call# ',I,text1
           enddo
501        close(41)

! get the flow rates for each run
          write(47,*)'**********************************************'
          write(47,*)'              FLOWRATES              '
          write(47,*)'**********************************************'
          write(47,102)'flowrt * * * * 0',(x_position(I),I=1,13)
        do I=1,9
                    write(string_1to9,'(i1)') I
                    file_name = 'HSFDieLipXFlowPerStep_run' // string_1to9 // '.res'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')

                    do J=1,MAXI
                              read(41,103,END=502) text1
                              write(47,104)'flowrt ','weight ',weight_number,'run#',I,' fcn_call# ',J,text1
                    enddo
502                 close(41)
          enddo
          do I=10,99
                    write(string_10to99,'(i2)') I
                    file_name = 'HSFDieLipXFlowPerStep_run' // string_10to99 // '.res'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')
                    do J=1,MAXI
                              read(41,103,END=503) text1
                              write(47,104)'flowrt ','weight ',weight_number,'run#',I,' fcn_call# ',J,text1
                    enddo
503                 close(41)
           enddo
           do I=100,nnoise
                    write(string_100to999,'(i3)') I
                    file_name = 'HSFDieLipXFlowPerStep_run' // string_100to999// '.res'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')
                    do J=1,MAXI
                              read(41,103,END=504) text1
                              write(47,104)'flowrt ','weight ',weight_number,'run#',I,' fcn_call# ',J,text1
                    enddo
504                 close(41)
           enddo
           close(47)

101 format (A2000)
102 format (A46,13E17.8)
103 format (A222)
104 format (A7,A7,A4,A5,I6,A11,I6,A2000)
105 format (A7,A4,A11,I6,A2000)
106 format (A46,A2000)
end
```

get_results.f90

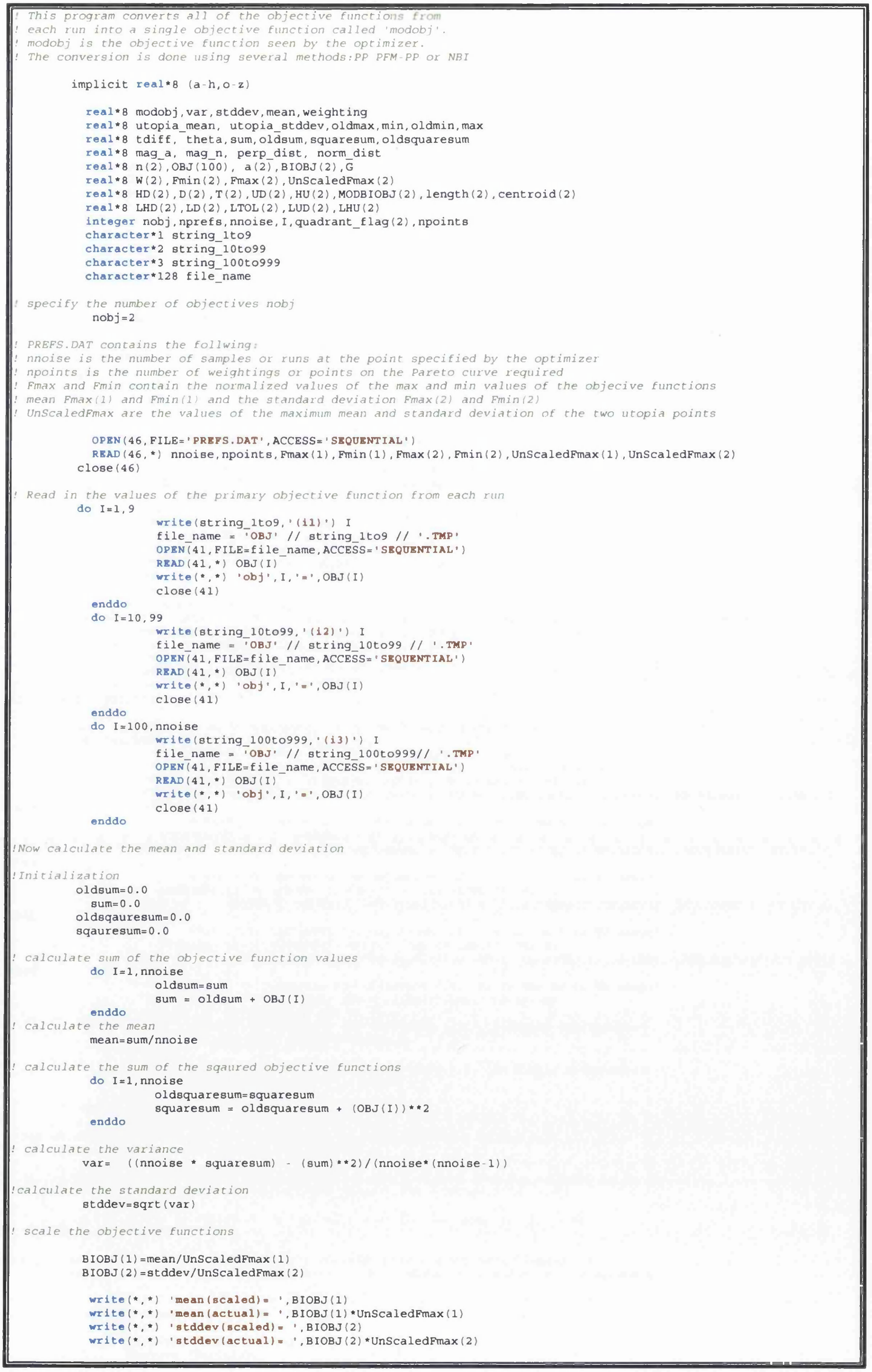
```
! This program is used to write the file: input_data.dat for each
! run which is used by makedatafiles.exe to make the
! datafiles for each run

          implicit real*8 (a-h,o-z)

          parameter (NVARMAX=8)
          parameter (NPOINTSMAX=100)
          real*8 Z(NVARMAX-3),flow,temp,nvalue,doe(NPOINTSMAX,NVARMAX)
          real*8 Zlb(NVARMAX),Zub(NVARMAX),nom_dv(NVARMAX),min_dv(NVARMAX)
          real*8 max_dv(NVARMAX),X(NVARMAX-3)
          integer I,J,NVAR,NPOINTS,counter
          character*80 text1,text2


! counter is the run number, run1 is the nominal run
          OPEN(95,FILE='counter.tmp',ACCESS='SEQUENTIAL')
          read(95,*)counter
          close(95)

! DOE.tp contains the number of points, variables, min max nom values of the parameters
! and the DOE matrix
          OPEN(96,FILE='DOE.tp',ACCESS='SEQUENTIAL')
          read(96,*) text1
          read(96,*) NVAR
          read(96,*) text1
          read(96,*) NPOINTS
          do I=1,NVAR
                    read(96,*) text1
                    read(96,*) nom_dv(i),min_dv(i),max_dv(i)
          enddo
          read(96,*) text1
          do I=1,NPOINTS
                              read(96,*) (doe(I,J),J=1,NVAR)
          enddo
          close(96)

! convert n, flow, temp
    do I=1,NVAR-5
                    Zlb(I)=min_dv(I)
                    Zub(I)=max_dv(I)
          enddo
! if the counter is zero then don't change parameters because this is
!the nominal run
          if(counter.eq.1) then
      nvalue = nom_dv(1)
                    flow= nom_dv(2)
                    temp = nom_dv(3)
          else
      nvalue = ((((doe(counter,1))-1)/(real(NPOINTS)-2))*(Zub(1)-Zlb(1)))+Zlb(1)
        flow = ((((doe(counter,2))-1)/(real(NPOINTS)-2))*(Zub(2)-Zlb(2)))+Zlb(2)
        temp = ((((doe(counter,3))-1)/(real(NPOINTS)-2))*(Zub(3)-Zlb(3)))+Zlb(3)
          endif
! finished converting the parameters n, flow, temp


! convert the lower and upper bounds of the shape variables
! in DoE.tp the min and max for the shape variables are min_dv(4 to 8)
! divide by 1000.0 because the bounds read in need to be in mm

          OPEN(95,FILE='DESVAR.TMP',ACCESS='SEQUENTIAL')
          READ (95,77)text1
          READ (95,77)text2
          do I=1,5
                    READ(95,*) Z(I)
          enddo
          CLOSE(95)

        do I=4,NVAR
                  J=I-3
                    Zlb(I)=Z(J)+min_dv(I)/1000.0
                    Zub(I)=Z(J)+max_dv(I)/1000.0
                    if(Zub(I).gt.3.2e-03)then
                              Zub(I)=3.2e-03
                    endif
                    if(Zlb(I).lt.0.1e-03)then
                              Zlb(I)=0.1e-03
                    endif
          enddo
! if the counter is zero then don't change shape variables because
! this is the nominal run
          if(counter.eq.1) then
                    do I=1,NVAR-3
                              X(I)= Z(I)
                    enddo
          else
          do I=4,NVAR
             J=I-3
             X(J)=((((doe(counter,I))-1)/(real(NPOINTS)-2))*(Zub(I)-Zlb(I)))+Zlb(I)
          enddo
          endif
! finished converting the shape variables

          OPEN(48,FILE='input_data.dat',ACCESS='SEQUENTIAL')

          write(48,*)'n='
          write(48,*) nvalue
          write(48,*)'flow='
          write(48,*) flow
          write(48,*)'temp='
          write(48,*) temp
          write(48,*)'x1='
          write(48,*) x(1)
          write(48,*)'x2='
          write(48,*) x(2)
          write(48,*)'x3='
          write(48,*) x(3)
          write(48,*)'x4='
          write(48,*) x(4)
          write(48,*)'x5='
          write(48,*) x(5)

          close(48)

          OPEN(95,FILE='counter.tmp',ACCESS='SEQUENTIAL')
          write(95,*)counter+1
          close(95)

 77       FORMAT (a80)
          end
```

input_data_maker.f90

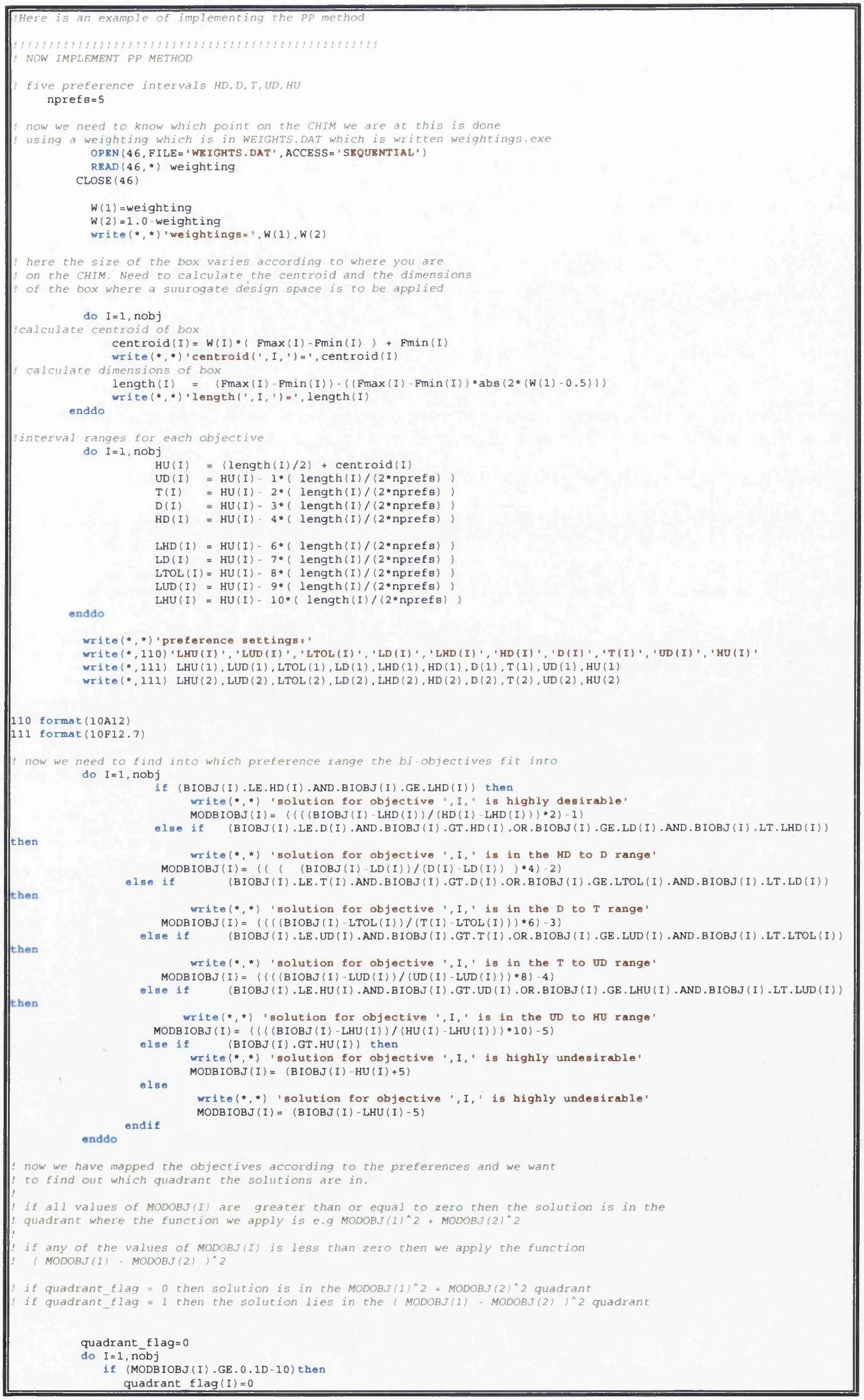


```
! This program writes the datafiles using input_data.dat which
! is made by input_data_maker.exe

        implicit real*8 (a-h,o-z)

          integer I,J,temp3
          real*8 n,flow,temperature,x(5),temp1,temp2
          character*80 text

          OPEN(47,FILE='input_data.dat',ACCESS='SEQUENTIAL')

          read(47,*) text
          read(47,*) n
          read(47,*) text
          read(47,*) flow
          read(47,*) text
          read(47,*) temperature
          read(47,*) text
          read(47,*) x(1)
          read(47,*) text
          read(47,*) x(2)
          read(47,*) text
          read(47,*) x(3)
          read(47,*) text
          read(47,*) x(4)
          read(47,*) text
          read(47,*) x(5)

          CLOSE(47)
! The NVALUE.DAT contains the value that power law index varies by
! the file is then read in slitsim.exe
          OPEN(47,FILE='NVALUE.DAT',ACCESS='SEQUENTIAL')
        write(47,*)n
          close(47)

! write the design variables to DESVAR.TMP
          OPEN(47,FILE='DESVAR.TMP',ACCESS='SEQUENTIAL')
        write(47,*)5
        write(47,*)0
          do I=1,5
                    write(47,*)x(I)
          enddo
          close(47)

! here example3.tmp is a copy of the original data file required by slitsim
! called example3.f_g, this is copied in the script.
! example3.f_g is re-created according to the new temperature and
! flow rates contained in input_data.dat
      OPEN(55,FILE='example3.tmp',ACCESS='SEQUENTIAL')
      OPEN(56,FILE='example3.f_g',ACCESS='SEQUENTIAL')
      DO J=1,12
              READ(55,1001) text
              write(56,1001) text
      END DO
      READ(55,1003) text
      write(56,1004) text,temperature
      DO J=1,3
              READ(55,1001) text
            write(56,1001) text
      END DO
      READ(55,1001) text
      write(56,1002) '1',flow/1000.0
      READ(55,1001) text
      write(56,1002) '2',flow/1000.0
      READ(55,1001) text
      write(56,1002) '3',flow/1000.0
      READ(55,1001) text
      write(56,1002) '4',flow/1000.0

      DO J=1,73
            READ(55,1001) text
              write(56,1001) text
      END DO
      close(55)
      close(56)

1001  format(a80)
1002  format(a5,E13.5)
1003  format(a14)
1004  format(a14,E11.4)

end
```

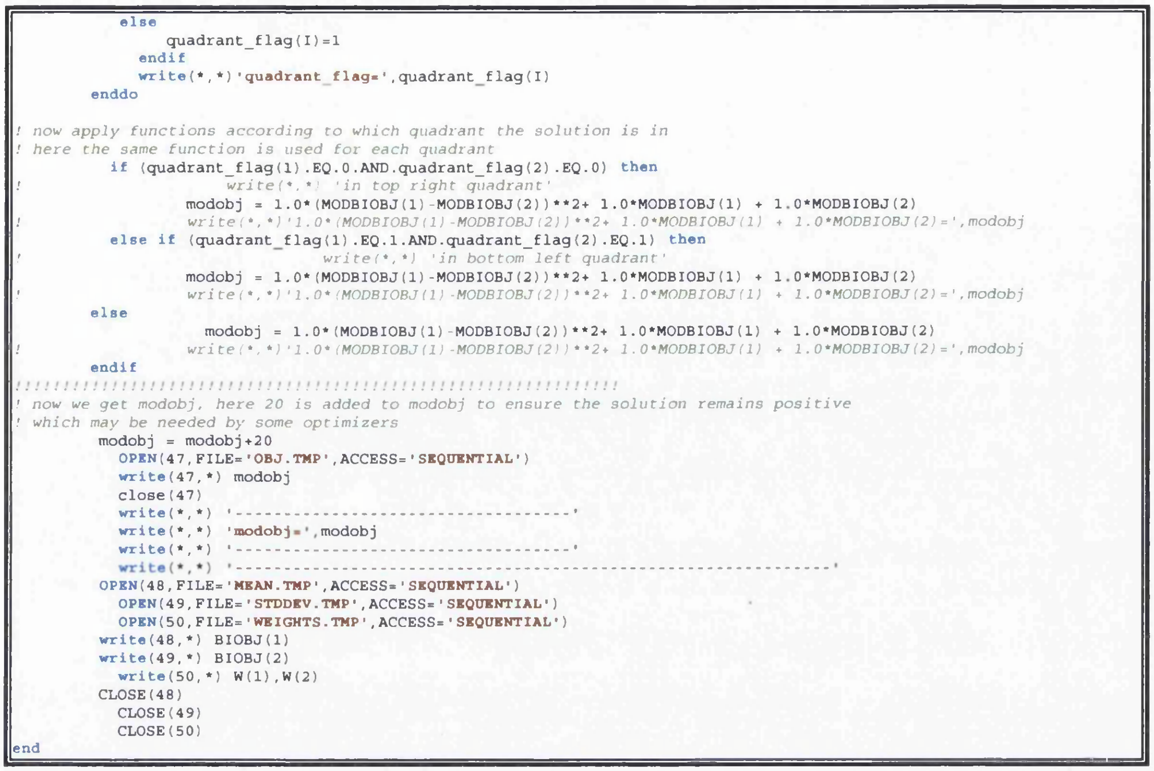
makedatafiles.f90



```
! This program converts all of the objective functions from
! each run into a single objective function called 'modobj'.
! modobj is the objective function seen by the optimizer.
! The conversion is done using several methods:PP PFM-PP or NBI

        implicit real*8 (a-h,o-z)

          real*8 modobj,var,stddev,mean,weighting
          real*8 utopia_mean, utopia_stddev,oldmax,min,oldmin,max
          real*8 tdiff, theta,sum,oldsum,squaresum,oldsquaresum
          real*8 mag_a, mag_n, perp_dist, norm_dist
          real*8 n(2),OBJ(100), a(2),BIOBJ(2),G
          real*8 W(2),Fmin(2),Fmax(2),UnScaledFmax(2)
          real*8 HD(2),D(2),T(2),UD(2),HU(2),MODBIOBJ(2),length(2),centroid(2)
          real*8 LHD(2),LD(2),LTOL(2),LUD(2),LHU(2)
          integer nobj,nprefs,nnoise,I,quadrant_flag(2),npoints
          character*1 string_1to9
          character*2 string_10to99
          character*3 string_100to999
          character*128 file_name

! specify the number of objectives nobj
           nobj=2

! PREFS.DAT contains the follwing:
! nnoise is the number of samples or runs at the point specified by the optimizer
! npoints is the number of weightings or points on the Pareto curve required
! Fmax and Fmin contain the normalized values of the max and min values of the objecive functions
! mean Fmax(1) and Fmin(1) and the standard deviation Fmax(2) and Fmin(2)
! UnScaledFmax are the values of the maximum mean and standard deviation of the two utopia points

           OPEN(46,FILE='PREFS.DAT',ACCESS='SEQUENTIAL')
           READ(46,*) nnoise,npoints,Fmax(1),Fmin(1),Fmax(2),Fmin(2),UnScaledFmax(1),UnScaledFmax(2)
         close(46)

! Read in the values of the primary objective function from each run
         do I=1,9
                    write(string_1to9,'(i1)') I
                    file_name = 'OBJ' // string_1to9 // '.TMP'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')
                    READ(41,*) OBJ(I)
                    write(*,*) 'obj',I,'=',OBJ(I)
                    close(41)
           enddo
           do I=10,99
                    write(string_10to99,'(i2)') I
                    file_name = 'OBJ' // string_10to99 // '.TMP'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')
                    READ(41,*) OBJ(I)
                    write(*,*) 'obj',I,'=',OBJ(I)
                    close(41)
           enddo
           do I=100,nnoise
                    write(string_100to999,'(i3)') I
                    file_name = 'OBJ' // string_100to999// '.TMP'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')
                    READ(41,*) OBJ(I)
                    write(*,*) 'obj',I,'=',OBJ(I)
                    close(41)
           enddo

!Now calculate the mean and standard deviation

!Initialization
         oldsum=0.0
           sum=0.0
         oldsqauresum=0.0
         sqauresum=0.0

! calculate sum of the objective function values
           do I=1,nnoise
                    oldsum=sum
                    sum = oldsum + OBJ(I)
           enddo
! calculate the mean
           mean=sum/nnoise

! calculate the sum of the sqaured objective functions
           do I=1,nnoise
                    oldsquaresum=squaresum
                    squaresum = oldsquaresum + (OBJ(I))**2
           enddo

! calculate the variance
          var=  ((nnoise * squaresum) - (sum)**2)/(nnoise*(nnoise-1))

!calculate the standard deviation
          stddev=sqrt(var)

! scale the objective functions

          BIOBJ(1)=mean/UnScaledFmax(1)
          BIOBJ(2)=stddev/UnScaledFmax(2)

           write(*,*) 'mean(scaled)= ',BIOBJ(1)
           write(*,*) 'mean(actual)= ',BIOBJ(1)*UnScaledFmax(1)
           write(*,*) 'stddev(scaled)= ',BIOBJ(2)
           write(*,*) 'stddev(actual)= ',BIOBJ(2)*UnScaledFmax(2)
```

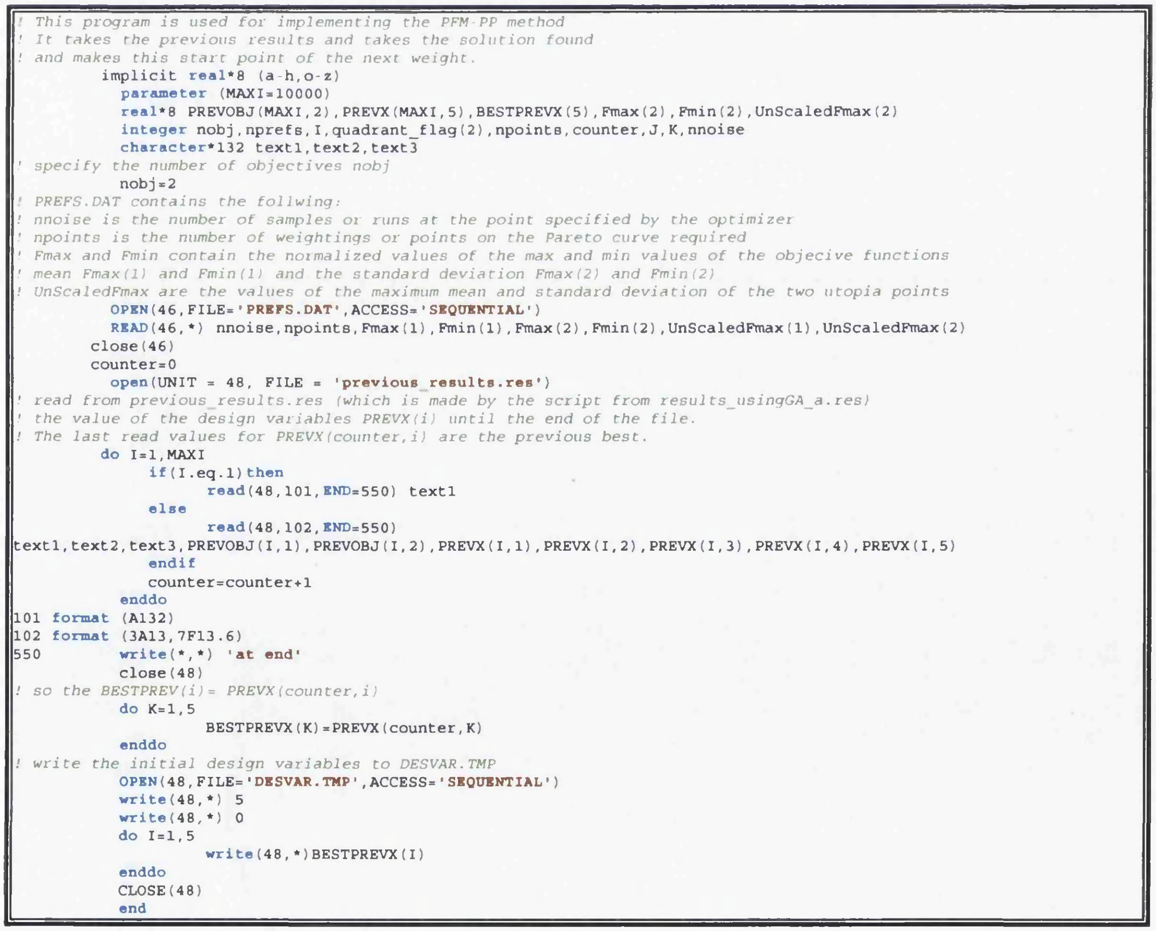


```
!Here is an example of implementing the PP method

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! NOW IMPLEMENT PP METHOD

! five preference intervals HD,D,T,UD,HU
     nprefs=5

! now we need to know which point on the CHIM we are at this is done
! using a weighting which is in WEIGHTS.DAT which is written weightings.exe
           OPEN(46,FILE='WEIGHTS.DAT',ACCESS='SEQUENTIAL')
           READ(46,*) weighting
         CLOSE(46)

           W(1)=weighting
           W(2)=1.0-weighting
           write(*,*)'weightings=',W(1),W(2)

! here the size of the box varies according to where you are
! on the CHIM. Need to calculate the centroid and the dimensions
! of the box where a suurogate design space is to be applied

          do I=1,nobj
!calculate centroid of box
              centroid(I)= W(I)*( Fmax(I)-Fmin(I) ) + Fmin(I)
              write(*,*)'centroid(',I,')=',centroid(I)
! calculate dimensions of box
              length(I)  =  (Fmax(I)-Fmin(I))-((Fmax(I)-Fmin(I))*abs(2*(W(1)-0.5)))
              write(*,*)'length(',I,')=',length(I)
        enddo

!interval ranges for each objective
          do I=1,nobj
                    HU(I)   = (length(I)/2) + centroid(I)
                    UD(I)   = HU(I)- 1*( length(I)/(2*nprefs) )
                    T(I)    = HU(I)- 2*( length(I)/(2*nprefs) )
                    D(I)    = HU(I)- 3*( length(I)/(2*nprefs) )
                    HD(I)   = HU(I)- 4*( length(I)/(2*nprefs) )

                    LHD(I) = HU(I)- 6*( length(I)/(2*nprefs) )
                    LD(I)  = HU(I)- 7*( length(I)/(2*nprefs) )
                    LTOL(I)= HU(I)- 8*( length(I)/(2*nprefs) )
                    LUD(I) = HU(I)- 9*( length(I)/(2*nprefs) )
                    LHU(I) = HU(I)- 10*( length(I)/(2*nprefs) )
        enddo

          write(*,*)'preference settings:'
          write(*,110)'LHU(I)','LUD(I)','LTOL(I)','LD(I)','LHD(I)','HD(I)','D(I)','T(I)','UD(I)','HU(I)'
          write(*,111) LHU(1),LUD(1),LTOL(1),LD(1),LHD(1),HD(1),D(1),T(1),UD(1),HU(1)
          write(*,111) LHU(2),LUD(2),LTOL(2),LD(2),LHD(2),HD(2),D(2),T(2),UD(2),HU(2)


110 format(10A12)
111 format(10F12.7)

! now we need to find into which preference range the bi-objectives fit into
          do I=1,nobj
                    if (BIOBJ(I).LE.HD(I).AND.BIOBJ(I).GE.LHD(I)) then
                         write(*,*) 'solution for objective ',I,' is highly desirable'
                         MODBIOBJ(I)= ((((BIOBJ(I)-LHD(I))/(HD(I)-LHD(I)))*2)-1)
                    else if   (BIOBJ(I).LE.D(I).AND.BIOBJ(I).GT.HD(I).OR.BIOBJ(I).GE.LD(I).AND.BIOBJ(I).LT.LHD(I))
then
                         write(*,*) 'solution for objective ',I,' is in the HD to D range'
                     MODBIOBJ(I)= (( (  (BIOBJ(I)-LD(I))/(D(I)-LD(I))  )*4)-2)
                並else if       (BIOBJ(I).LE.T(I).AND.BIOBJ(I).GT.D(I).OR.BIOBJ(I).GE.LTOL(I).AND.BIOBJ(I).LT.LD(I))
then
                         write(*,*) 'solution for objective ',I,' is in the D to T range'
                     MODBIOBJ(I)= ((((BIOBJ(I)-LTOL(I))/(T(I)-LTOL(I)))*6)-3)
                  else if     (BIOBJ(I).LE.UD(I).AND.BIOBJ(I).GT.T(I).OR.BIOBJ(I).GE.LUD(I).AND.BIOBJ(I).LT.LTOL(I))
then
                         write(*,*) 'solution for objective ',I,' is in the T to UD range'
                     MODBIOBJ(I)= ((((BIOBJ(I)-LUD(I))/(UD(I)-LUD(I)))*8)-4)
                  else if     (BIOBJ(I).LE.HU(I).AND.BIOBJ(I).GT.UD(I).OR.BIOBJ(I).GE.LHU(I).AND.BIOBJ(I).LT.LUD(I))
then
                        write(*,*) 'solution for objective ',I,' is in the UD to HU range'
                    MODBIOBJ(I)= ((((BIOBJ(I)-LHU(I))/(HU(I)-LHU(I)))*10)-5)
                  else if     (BIOBJ(I).GT.HU(I)) then
                         write(*,*) 'solution for objective ',I,' is highly undesirable'
                         MODBIOBJ(I)= (BIOBJ(I)-HU(I)+5)
                  else
                          write(*,*) 'solution for objective ',I,' is highly undesirable'
                          MODBIOBJ(I)= (BIOBJ(I)-LHU(I)-5)
                endif
          enddo

! now we have mapped the objectives according to the preferences and we want
! to find out which quadrant the solutions are in.
!
! if all values of MODOBJ(I) are  greater than or equal to zero then the solution is in the
! quadrant where the function we apply is e.g MODOBJ(1)^2 + MODOBJ(2)^2
!
! if any of the values of MODOBJ(I) is less than zero then we apply the function
!   ( MODOBJ(1)  - MODOBJ(2)  )^2

! if quadrant_flag = 0 then solution is in the MODOBJ(1)^2 + MODOBJ(2)^2 quadrant
! if quadrant_flag = 1 then the solution lies in the ( MODOBJ(1)  - MODOBJ(2)  )^2 quadrant


          quadrant_flag=0
          do I=1,nobj
             if (MODBIOBJ(I).GE.0.1D-10)then
                quadrant_flag(I)=0
```



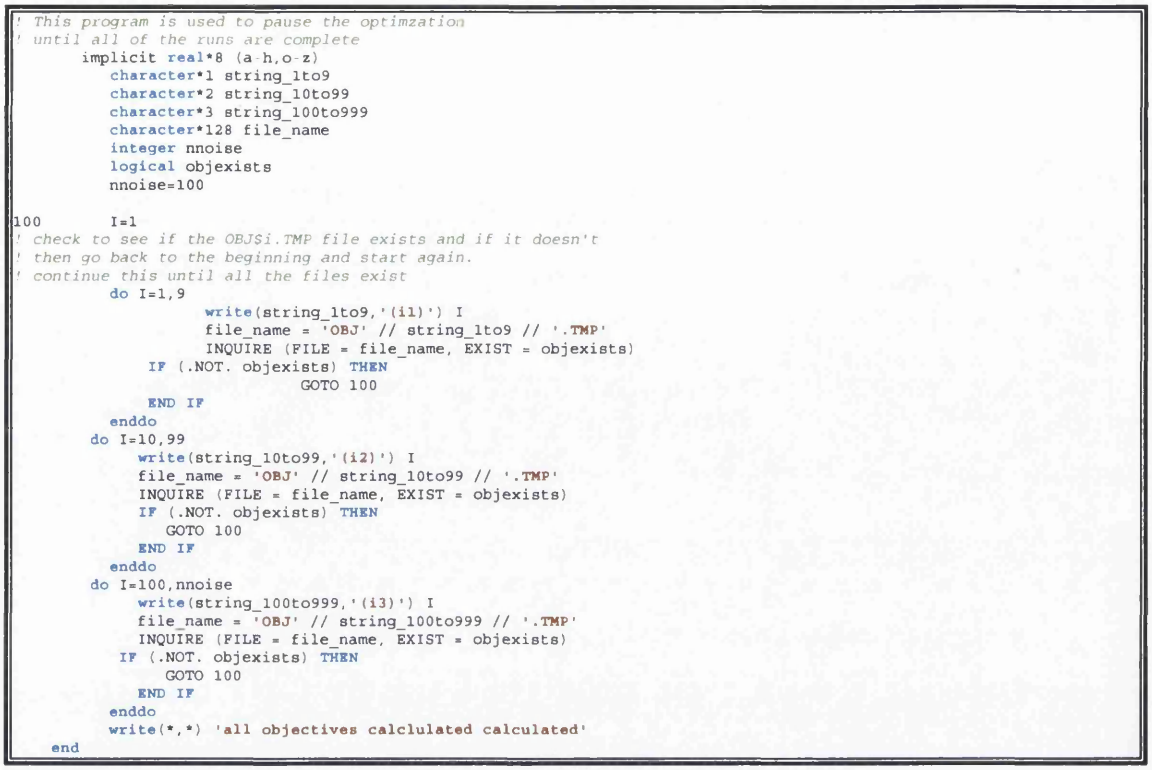
```
           else
                quadrant_flag(I)=1
             endif
             write(*,*)'quadrant_flag=',quadrant_flag(I)
        enddo

! now apply functions according to which quadrant the solution is in
! here the same function is used for each quadrant
          if (quadrant_flag(1).EQ.0.AND.quadrant_flag(2).EQ.0) then
!                     write(*,*) 'in top right quadrant'
                  modobj = 1.0*(MODBIOBJ(1)-MODBIOBJ(2))**2+ 1.0*MODBIOBJ(1) + 1.0*MODBIOBJ(2)
!                 write(*,*)'1.0*(MODBIOBJ(1)-MODBIOBJ(2))**2+ 1.0*MODBIOBJ(1) + 1.0*MODBIOBJ(2)=',modobj
          else if (quadrant_flag(1).EQ.1.AND.quadrant_flag(2).EQ.1) then
!                               write(*,*) 'in bottom left quadrant'
                  modobj = 1.0*(MODBIOBJ(1)-MODBIOBJ(2))**2+ 1.0*MODBIOBJ(1) + 1.0*MODBIOBJ(2)
!                 write(*,*)'1.0*(MODBIOBJ(1)-MODBIOBJ(2))**2+ 1.0*MODBIOBJ(1) + 1.0*MODBIOBJ(2)=',modobj
        else
                    modobj = 1.0*(MODBIOBJ(1)-MODBIOBJ(2))**2+ 1.0*MODBIOBJ(1) + 1.0*MODBIOBJ(2)
!                 write(*,*)'1.0*(MODBIOBJ(1)-MODBIOBJ(2))**2+ 1.0*MODBIOBJ(1) + 1.0*MODBIOBJ(2)=',modobj
        endif
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! now we get modobj, here 20 is added to modobj to ensure the solution remains positive
! which may be needed by some optimizers
         modobj = modobj+20
           OPEN(47,FILE='OBJ.TMP',ACCESS='SEQUENTIAL')
           write(47,*) modobj
           close(47)
           write(*,*) '-----------------------------------'
           write(*,*) 'modobj=',modobj
           write(*,*) '-----------------------------------'
           write(*,*) '-------------------------------------------------------------'
         OPEN(48,FILE='MEAN.TMP',ACCESS='SEQUENTIAL')
           OPEN(49,FILE='STDDEV.TMP',ACCESS='SEQUENTIAL')
           OPEN(50,FILE='WEIGHTS.TMP',ACCESS='SEQUENTIAL')
         write(48,*) BIOBJ(1)
         write(49,*) BIOBJ(2)
           write(50,*) W(1),W(2)
         CLOSE(48)
           CLOSE(49)
           CLOSE(50)
end
```

obj2modobj.f90



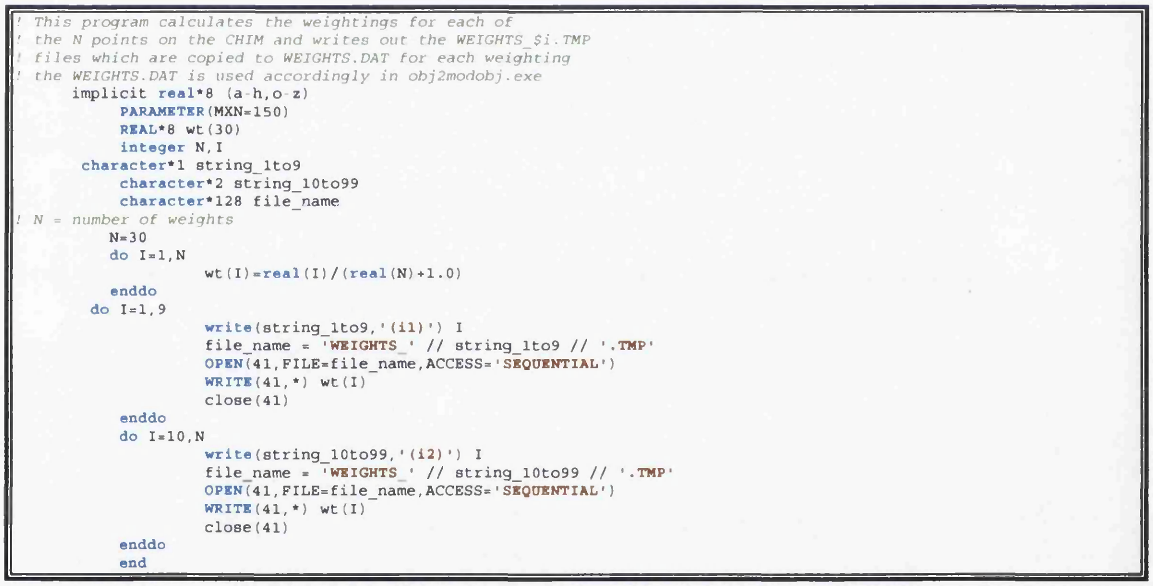
```
! This program is used for implementing the PFM-PP method
! It takes the previous results and takes the solution found
! and makes this start point of the next weight.
         implicit real*8 (a-h,o-z)
           parameter (MAXI=10000)
           real*8 PREVOBJ(MAXI,2),PREVX(MAXI,5),BESTPREVX(5),Fmax(2),Fmin(2),UnScaledFmax(2)
           integer nobj,nprefs,I,quadrant_flag(2),npoints,counter,J,K,nnoise
           character*132 text1,text2,text3
! specify the number of objectives nobj
           nobj=2
! PREFS.DAT contains the follwing:
! nnoise is the number of samples or runs at the point specified by the optimizer
! npoints is the number of weightings or points on the Pareto curve required
! Fmax and Fmin contain the normalized values of the max and min values of the objecive functions
! mean Fmax(1) and Fmin(1) and the standard deviation Fmax(2) and Fmin(2)
! UnScaledFmax are the values of the maximum mean and standard deviation of the two utopia points
          OPEN(46,FILE='PREFS.DAT',ACCESS='SEQUENTIAL')
          READ(46,*) nnoise,npoints,Fmax(1),Fmin(1),Fmax(2),Fmin(2),UnScaledFmax(1),UnScaledFmax(2)
        close(46)
        counter=0
          open(UNIT = 48, FILE = 'previous_results.res')
! read from previous_results.res (which is made by the script from results_usingGA_a.res)
! the value of the design variables PREVX(i) until the end of the file.
! The last read values for PREVX(counter,i) are the previous best.
         do I=1,MAXI
              if(I.eq.1)then
                    read(48,101,END=550) text1
              else
                    read(48,102,END=550)
text1,text2,text3,PREVOBJ(I,1),PREVOBJ(I,2),PREVX(I,1),PREVX(I,2),PREVX(I,3),PREVX(I,4),PREVX(I,5)
              endif
              counter=counter+1
           enddo
101 format (A132)
102 format (3A13,7F13.6)
550        write(*,*) 'at end'
           close(48)
! so the BESTPREV(i)= PREVX(counter,i)
           do K=1,5
                    BESTPREVX(K)=PREVX(counter,K)
           enddo
! write the initial design variables to DESVAR.TMP
           OPEN(48,FILE='DESVAR.TMP',ACCESS='SEQUENTIAL')
           write(48,*) 5
           write(48,*) 0
           do I=1,5
                    write(48,*)BESTPREVX(I)
           enddo
           CLOSE(48)
           end
```

previousbest.f90



```
! This program is used to pause the optimzation
! until all of the runs are complete
       implicit real*8 (a-h,o-z)
          character*1 string_1to9
          character*2 string_10to99
          character*3 string_100to999
          character*128 file_name
          integer nnoise
          logical objexists
          nnoise=100

100       I=1
! check to see if the OBJ$i.TMP file exists and if it doesn't
! then go back to the beginning and start again.
! continue this until all the files exist
          do I=1,9
                    write(string_1to9,'(i1)') I
                    file_name = 'OBJ' // string_1to9 // '.TMP'
                    INQUIRE (FILE = file_name, EXIST = objexists)
              IF (.NOT. objexists) THEN
                              GOTO 100
              END IF
          enddo
        do I=10,99
             write(string_10to99,'(i2)') I
             file_name = 'OBJ' // string_10to99 // '.TMP'
             INQUIRE (FILE = file_name, EXIST = objexists)
             IF (.NOT. objexists) THEN
                GOTO 100
             END IF
          enddo
        do I=100,nnoise
             write(string_100to999,'(i3)') I
             file_name = 'OBJ' // string_100to999 // '.TMP'
             INQUIRE (FILE = file_name, EXIST = objexists)
           IF (.NOT. objexists) THEN
                GOTO 100
             END IF
          enddo
          write(*,*) 'all objectives calclulated calculated'
    end
```

wait.for



```
! This program calculates the weightings for each of
! the N points on the CHIM and writes out the WEIGHTS_$i.TMP
! files which are copied to WEIGHTS.DAT for each weighting
! the WEIGHTS.DAT is used accordingly in obj2modobj.exe
      implicit real*8 (a-h,o-z)
           PARAMETER(MXN=150)
           REAL*8 wt(30)
           integer N,I
       character*1 string_1to9
           character*2 string_10to99
           character*128 file_name
! N = number of weights
          N=30
          do I=1,N
                    wt(I)=real(I)/(real(N)+1.0)
          enddo
        do I=1,9
                    write(string_1to9,'(i1)') I
                    file_name = 'WEIGHTS_' // string_1to9 // '.TMP'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')
                    WRITE(41,*) wt(I)
                    close(41)
           enddo
           do I=10,N
                    write(string_10to99,'(i2)') I
                    file_name = 'WEIGHTS_' // string_10to99 // '.TMP'
                    OPEN(41,FILE=file_name,ACCESS='SEQUENTIAL')
                    WRITE(41,*) wt(I)
                    close(41)
           enddo
           end
```

weightings.f90

A2.3 Explanation of Figures 5.23 and 5.24

Figure A2.5 shows the raw data results for a sample of size 100 taken about the nominal choker bar result for weight 1. The centre of each plot represents a value of zero for the primary objective function (Equation 5.8) and the outer boundary is a value of 10, the radial co-ordinate is the sample number. Figure A2.6 shows the same data combined into a histogram.

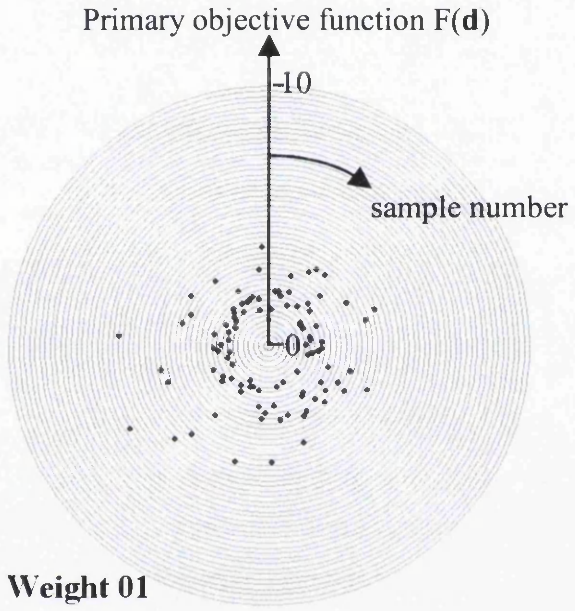


Figure A2.5: Performance Scatter plot for weight 1 Pareto solution in Figure 5.25

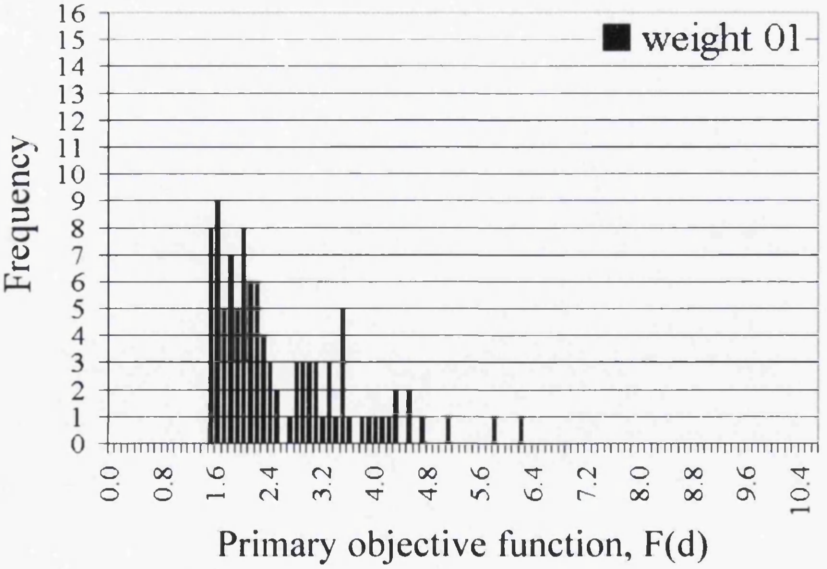


Figure A2.6: Histogram showing the performance variation for weight 1 Pareto solution in Figure 5.25

LIST OF PUBLICATIONS

Journal Papers

- Bates S.J., Sienz J., Pittman J.F.T., "Robust Design Optimization of Extrusion Slit Die Design using the Pareto Front Marching Method based on Physical Programming", invited for publication and submitted to AIAA journal, 2003.
- Bates S.J., Sienz J., Pittman J.F.T., "Comparison of a gradient-free, a gradient-based and a mid-range approximation optimization method applied to slit die design", submitted to Journal Computers and Structures, 2003.
- Bates S. J., Sienz J., "Formulation of the Audze–Eglais Uniform Latin Hypercube design of experiments", Journal. Advances in Engineering Software, vol. 34/8 pp. 493 – 506, 2003.
- Bates S.J., Pittman J.F.T., Sienz J., Langley D.S. "Enhancing Slit Die Performance by Optimization of Restrictor Profiles", Journal for Polymer Engineering and Science, August 2003.

Conference Papers

- Bates S.J., Sienz J., Toropov V.V., "Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm", 45th AIAA Conference, Palm Springs, California, April 2004.
- Bates S.J., Sienz J., Pittman J.F.T., "Robust Design Optimization of Extrusion Slit Die Design Using Several Multi-Objective Optimization Methods", proc. 5th WCSMO conf., Venice, Italy, 2003.
- Bates S.J., Randle J., Gambling M., Jolly M.R., "A holistic virtual design process applied to an a-pillar subjected to roof crush", proc. 3rd Altair UK Hyperworks conf., Gaydon, UK, 2002.
- Bates S.J., Sienz J., Pittman J.F.T., Langley D.S. "Application of Robust Design Optimization to Extrusion Slit Die Design", proc. 3rd Int. conf. on Engineering Computational Technology, Prague, 2002, ISBN: 0-948749-86-5
- Bates S.J., Sienz J., Pittman J.F.T. "Comparison of conventional optimization with robust design techniques for slit die design", Proc. 4th ASMO UK / ISSMO conf., Newcastle, UK, 2002, ISBN: 0-7017-0137-4
- Sienz J., Pittman J.F.T., Ettinger H., Bates S.J., "Optimization of Thermoplastics Extrusion Die Design: Parameterization, Algorithms and Strategies", 5th Int. ESAFORM conf. 2002.
- Sienz J., Bates S.J., Langley D.S., Pittman J.F.T. "Comparison of globally converging optimization with mid-range approximation based optimization applied to slit die design" Proc. 3rd ASMO UK / ISSMO conf., Harrogate, UK, 2001,ISBN: 0-85316-219-0.