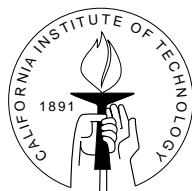


# Evaluating Imprecision in Engineering Design

Thesis by  
William S. Law

In Partial Fulfillment of the Requirements  
for the Degree of  
Doctor of Philosophy



California Institute of Technology  
Pasadena, California

1996

(Defended May 23, 1996)

© 1996

William S. Law

All rights Reserved

## Acknowledgements

I have often wondered why a graduate student's papers always bear his (or her) advisor's name as a co-author, yet a thesis bears only the student's name. Perhaps it is so the student may properly acknowledge his advisor's contribution. If I had to attribute every idea that directly or indirectly resulted from discussions with my advisor, Prof. Erik Antonsson, there would be a footnote on every other page of this thesis. I came to Caltech largely because of my initial impression of Prof. Antonsson. He has given me no reason to regret that decision.

The work presented in this thesis has been supported, in part, by the National Science Foundation under NSF Grant Numbers DDM-9201424 and DMI-9523232. Any opinions, findings, conclusions or recommendations expressed in this publication are my own and do not necessarily reflect the views of the sponsors.

I am indebted to the managers and engineers at General Electric Aircraft Engines and Corporate Research and Development, an anonymous major U.S. automobile manufacturer, Alcoa Technical Center, and Amerigon Incorporated, who were so generous with their time. My education and this thesis would have been incomplete without them. I am grateful to my office-mate and colleague Michael Scott for his contributions to this work, in particular the family of aggregation functions discussed in Section 2.5 and the finite element model used in Section 4.2. Jim Radford assisted in writing Perl programs for handling various files. Andrew Lewis has been a valuable source of technical advice and  $\text{\LaTeX}$  packages. His comments have greatly improved this thesis. My friends in the Thomas building, at Caltech, and beyond have made it all worthwhile, especially Howie Choset, Michael and Donna Vanik, Raffaello D'Andrea, Sharon and Kent Wong, Karen and Alan Lo, and the Thomas lunch group. My relatives and family friends in the Los Angeles area not only fed me, but gave me a sense of belonging. To my brothers, thank you for putting up with me.

I dedicate this thesis to my best friend, Mei-Hsia Tan, for providing a good reason to graduate and for being, and to my mother and father, for being my parents and for providing.

# Evaluating Imprecision in Engineering Design

by

William S. Law

In Partial Fulfillment of the  
Requirements for the Degree of  
Doctor of Philosophy

## Abstract

*Imprecision* is uncertainty that arises because of vague or incomplete information. Preliminary design information is characteristically imprecise: specifications and requirements are subject to change, and the design description is vague and incomplete. Yet many powerful evaluation tools, including finite element models, expect precisely specified data. Thus it is common for engineers to evaluate promising designs one by one. Alternatively, optimization may be used to search for the single “best” design. These approaches focus on individual, precisely specified points in the design space and provide limited information about the full range of acceptable designs.

An alternative approach would be to evaluate *sets* of designs. The *method of imprecision* uses the mathematics of fuzzy sets in order to represent imprecision as preferences among designs:

- Functional requirements model the customer’s direct preference on performance variables based on *performance considerations*: the quantified aspects of design performance represented by performance variables.
- Design preferences model the customer’s anticipated preference on design variables based on *design considerations*: the unquantified aspects of design performance not represented by performance variables.

Design preferences provide a formal structure for representing “soft” issues such as aesthetics and manufacturability and quantifying their consequences.

This thesis describes continuing work in bringing the method of imprecision closer to implementation as a decision-making methodology for engineering design. The two principal contributions of this work are a clearer interpretation of the elements that comprise the method and a more efficient computational implementation.

The proposed method for modeling design decisions in the presence of imprecision is defined in detail. The decision-maker is modeled as a hierarchy of preference aggregation operations. Axioms for rational design decision-making are used to define aggregation operations that are suitable for design. An electric vehicle design example illustrates the method. In particular, the process of determining preferences and a preference aggregation hierarchy is shown to be both feasible and informative. Efficient computational methods for performing preference calculations are introduced. These methods use experiment design to explore the design space and optimization assisted by linear approximation to map preferences. A user-specified fractional precision allows the number of function evaluations to be traded-off against the quality of the answer obtained. The computational methods developed are verified on design problems from aircraft engine development and automobile body design. Procedures for specifying preferences and group decision-making are described. These procedures provide not only a pragmatic interpretation of the method, but also an informal solution to the problem of bargaining: prerequisites for bringing the method to design problems in the real world.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Organization of Thesis . . . . .	4
<b>2</b>	<b>Modeling Imprecision in Design</b>	<b>7</b>
2.1	Basic Definitions . . . . .	7
2.2	Representing Imprecision . . . . .	9
2.3	Aggregating Preferences . . . . .	12
2.4	Weights . . . . .	17
2.5	Weighted Means . . . . .	19
2.6	Hierarchical Weighted Design . . . . .	23
2.7	Example: Design of an Electric Vehicle . . . . .	26
2.8	Related Work . . . . .	33
2.9	Conclusions . . . . .	42
<b>3</b>	<b>Calculating Imprecision</b>	<b>46</b>
3.1	The Level Interval Algorithm . . . . .	47
3.2	Optimization . . . . .	50
3.3	Mapping Design Imprecision . . . . .	52
3.4	Design of Experiments . . . . .	63
3.5	Conclusions . . . . .	71
<b>4</b>	<b>Computational Implementation</b>	<b>76</b>
4.1	The Engine Development Cost Estimator . . . . .	76
4.1.1	Example: Development of a Turbofan Engine . . . . .	80

4.2	Vehicle Structure Design . . . . .	90
4.2.1	Example: Finite Element Analysis of a Vehicle Body . . . . .	94
4.2.2	Example: Quadratic Approximations to the Finite Element Analysis . . . . .	103
4.3	Conclusions . . . . .	109
<b>5</b>	<b>Interpreting Imprecision</b>	<b>114</b>
5.1	Specifying Preferences . . . . .	114
5.2	A Scenario for Implementation . . . . .	121
5.3	Conclusions . . . . .	130
<b>6</b>	<b>Conclusions</b>	<b>136</b>
<b>A</b>	<b>Approximation by Tiling</b>	<b>145</b>

## List of Figures

1.1	Evaluating individual designs one by one. . . . .	2
1.2	Evaluating sets of designs. . . . .	2
2.1	Imprecise functional requirement “less than 8 seconds”. . . . .	10
2.2	Amerigon’s electric vehicle chassis © <i>Copyright 1996 CALSTART, Inc., all rights reserved. Permission to use granted April 30, 1996.</i> . . . . .	26
2.3	Design preference aggregation hierarchy for an electric vehicle design. . . . .	27
2.4	B pillar design variables. . . . .	28
3.1	Discretized design preference $\mu_{d_1}$ . . . . .	48
3.2	The Level Interval Algorithm. . . . .	49
3.3	Design preference intervals for $d_1$ and $d_2$ . . . . .	52
3.4	Combined design preference $\mu_d = \mathcal{P}(\mu_{d_1}, \mu_{d_2})$ . $\alpha_{i,j} = \mathcal{P}(\alpha_i, \alpha_j)$ . . . . .	53
3.5	Combined design preference $\mu_d = \mathcal{P}_{\min}(\mu_{d_1}, \mu_{d_2})$ . . . . .	53
3.6	Combined design preference $\mu_d = \mathcal{P}_{\Pi}(\mu_{d_1}, \mu_{d_2})$ . . . . .	54
3.7	Reducing the search space with linear regression. . . . .	57
3.8	Approximated $\alpha$ -cut $P_\epsilon^{d'}$ on the PVS. . . . .	60
3.9	Number of evaluations for a fractional factorial design. . . . .	65
3.10	Points evaluated in a central composite design. . . . .	66
3.11	Number of evaluations for a central composite design. . . . .	67
3.12	Minimizing $\int_{-L_1}^{L_1} E^2 dx_1$ with offset $\Delta_j$ ( $f_j = K_{11}x_1^2$ ). . . . .	69
4.1	Schematic diagram of a turbofan engine. . . . .	77
4.2	The Imprecise Design Tool and the EDCE. . . . .	78
4.3	Design preferences $\mu_{d_1}, \dots, \mu_{d_8}$ for the turbojet option. . . . .	82

4.4	Design preferences $\mu_{d_1}, \dots, \mu_{d_8}$ for the turbofan option. . . . .	83
4.5	$\mu_o, \mu_d$ , and $\mu_p$ on the PVS, for augmenting the turbojet. . . . .	84
4.6	$\mu_o, \mu_d$ , and $\mu_p$ on the PVS, for modifying the turbofan. . . . .	85
4.7	$\mu_d$ on the PVS for the turbojet and turbofan options. . . . .	88
4.8	The body-in-white. . . . .	91
4.9	The Imprecise Design Tool. . . . .	93
4.10	Finite element model of a car body. . . . .	94
4.11	Quadratic bending stiffness. . . . .	104
4.12	Quadratic torsional stiffness. . . . .	105
4.13	The $\alpha$ -cut $D_\epsilon^d$ . . . . .	108
4.14	$D_\epsilon^d$ mapped onto $P_\epsilon^d$ and $P_\epsilon^{d'}$ . . . . .	108
5.1	Aggregating three $\alpha$ -cuts: level sets for $\mu_d = \mathcal{P}_\Pi(\mu_{d_1}, \mu_{d_2})$ . . . . .	119
5.2	Aggregating three $\alpha$ -cuts: level sets for $\mu_d = \mathcal{P}_{\max'}(\mu_{d_1}, \mu_{d_2})$ . . . . .	119
5.3	Functional requirement aggregation hierarchy for an electric vehicle design. . . . .	125
5.4	Aggregating three $\alpha$ -cuts with weights, $\omega_1 > \omega_2$ . . . . .	127
5.5	Overall preference aggregation hierarchy. . . . .	128
A.1	Tiling approximation for $K_B$ . . . . .	146
A.2	Misaligned tiles. . . . .	147

## List of Tables

2.1	Design-appropriate aggregation functions based on the weighted means, parameterized in $s$ and $c$ . . . . .	23
4.1	Linear regression results for bending stiffness $K_B$ . . . . .	96
4.2	Linear regression results for torsional stiffness $K_T$ . . . . .	98
4.3	Center point offsets for bending and torsional stiffness. . . . .	99
4.4	Extrema for bending and torsional stiffness in $D_c^d$ . . . . .	100
4.5	Number of function evaluations required by the IDT. . . . .	101
4.6	Linear regression results for quadratic $K_B$ and $K_T$ . . . . .	106
4.7	Extrema for quadratic $K_B$ and $K_T$ . . . . .	106
4.8	Approximation errors for quadratic $K_B$ and $K_T$ . . . . .	107

## List of Symbols

Symbol	: Description and page when applicable
$\mathbf{A}$	: matrix of coefficients $a_{ji}$ in linear approximation $\vec{f}'$ , 55
$d_i$	: $i$ th design variable ( $i = 1, \dots, n$ ), 8
$[d_{i_{\min}}^{\alpha_k}, d_{i_{\max}}^{\alpha_k}]$	: individual design preference interval at $\alpha_k$ ( $k = 1, \dots, M$ ), 48
$\vec{d}$	: vector of design variables $[d_1, \dots, d_n]^T$ , 8
$\vec{d}_h^{\square}$	: corner point evaluated in a fractional factorial experiment ( $h = 1, \dots, N^{\square}$ ), 69
$\vec{d}_i^{\pm}$	: pair of points evaluated in the $\pm d_i$ directions for a one-factor-at-a-time experiment, 69
$\vec{d}_{\text{ctr}}$	: center point of $D_{\epsilon}^d$ , 55
$D_{\alpha_k}^d$	: combined design preference $\alpha$ -cut set on the DVS at $\alpha_k$ ( $k = 1, \dots, M$ ), 49
$\Delta_j$	: constant offset at $\vec{d}_{\text{ctr}}$ in linear approximation $f'_j$ , 55
$D_{\alpha_k}^p$	: combined functional requirement $\alpha$ -cut set mapped onto the DVS at $\alpha_k$ ( $k = 1, \dots, M$ ), 58
$D_{\alpha_k}^{p'}$	: approximation to $D_{\alpha_k}^p$ , obtained using $\vec{f}'$ , 58
$E(\vec{x})$	: approximation error at $\vec{d} = \vec{d}_{\text{ctr}} + \vec{x}$ , 69
$\epsilon$	: infinitesimal preference ( $0 < \epsilon \ll 1$ ), 55
$f'_j$	: (linear) approximation to $f_j$ , 55
$\vec{f}'$	: (linear) approximation to $\vec{f}$ , 55
$K_B$	: bending stiffness (lbf/in), 95
$K_T$	: torsional stiffness (ft-lbf/ $^{\circ}$ ), 95

$\kappa_{ji}^{\alpha_k}$	: design sensitivity of $p_j$ to $[d_{i\min}^{\alpha_k}, d_{i\max}^{\alpha_k}]$ normalized for each $p_j$ at $\alpha_1$ , 57
$\mathcal{L}$	: set of acceptably linear design variables, 70
$M_D$	: number of discrete preference levels for the combined design preference ( $M_D \geq M_I$ ), 114
$M_I$	: number of discrete preference levels for individual design preferences and functional requirements, 114
$M_P$	: number of discrete preference levels for the combined functional requirement ( $M_P \geq M_I$ ), 114
$\mu_{d_i}$	: individual design preference for $d_i$ ( $i = 1, \dots, n$ ), 11
$\mu_d(\vec{d})$	: combined design preference on the DVS, 47
$\mu_d(\vec{p})$	: combined design preference mapped onto the PVS, 47
$\mu_o^*$	: peak overall preference, achieved by designs $\vec{d} \in \mathcal{X}^*$ , 12
$\mu_o(\vec{d})$	: overall preference on the DVS, 12
$\mu_o(\vec{p})$	: overall preference on the PVS, 47
$\mu_{p_j}$	: individual functional requirement for $p_j$ ( $j = 1, \dots, q$ ), 10
$\mu_p(\vec{d})$	: combined functional requirement mapped onto the DVS, 47
$\mu_p(\vec{p})$	: combined functional requirement on the PVS, 47
$N^\square$	: number of points evaluated in a fractional factorial experiment, 69
$n_{\mathcal{L}}$	: number of acceptably linear design variables, 70
$p_j$	: $j$ th performance variable, $p_j = f_j(\vec{d})$ ( $j = 1, \dots, q$ ), 9
$[p_{j\min}^{\alpha_k}, p_{j\max}^{\alpha_k}]$	: individual functional requirement interval at $\alpha_k$ ( $k = 1, \dots, M$ ), 49
$\vec{p}$	: vector of performance variables $[p_1, \dots, p_q]^T = \vec{f}(\vec{d})$ , 9
$P_{\alpha_k}^d$	: approximation to $P_{\alpha_k}^d$ , obtained through $\vec{f}'$ , 59
$P_{\alpha_k}^{d^\square}$	: $q$ -cube approximation to $P_{\alpha_k}^d$ , 55
$\mathcal{P}$	: aggregation function which trades-off preferences, 12

$\mathcal{P}_c$	: full class of design appropriate aggregation functions reparameterized in $c$ ( $\mathcal{P}_{s \leq 0}$ and $\mathcal{P}_{s > 0'}$ ), 22
$\hat{\mathcal{P}}$	: aggregation operation ( $\mathcal{P}^\mu, \mathcal{P}^\omega$ ) which trades-off $\hat{\mu} = (\mu, \omega)$ pairs, 23
$\mathcal{P}_{\max'}$	: maximal design appropriate aggregation function, 21
$\mathcal{P}_{\min}$	: non-compensating aggregation function, 15
$\mathcal{P}_{\Pi}$	: fully compensating aggregation function, 16
$\mathcal{P}_s$	: class of weighted root-mean-power aggregation functions, parameterized in $s$ ( $-\infty \leq s \leq \infty$ ), 20
$\mathcal{P}_{s \leq 0}$	: design appropriate subclass of non- to fully compensating weighted root-mean-power aggregation functions, 20
$\mathcal{P}_{s > 0'}$	: class of design appropriate super- to maximally compensating aggregation functions ( $\mathcal{P}_{c > 1}$ ), 21
$P_{\alpha_k}^d$	: combined design preference $\alpha$ -cut set mapped onto the PVS at $\alpha_k$ ( $k = 1, \dots, M$ ), 55
$P_{\alpha_k}^p$	: combined functional requirement $\alpha$ -cut set on the PVS at $\alpha_k$ ( $k = 1, \dots, M$ ), 58
$\mathcal{X}_i$	: set of valid values for the design variable $d_i$ , 8
$\mathcal{X}^*$	: peak preference designs, $\forall \vec{d} \in \mathcal{X}^* \mu_o(\vec{d}) = \mu_o^*$ , 12
$\mathcal{Y}_j$	: set of valid values for the performance variable $p_j$ , 9
$\mathcal{Y}^*$	: peak preference performances, $\forall \vec{p} \in \mathcal{Y}^* = \vec{f}(\mathcal{X})^* \mu_o(\vec{p}) = \mu_o^*$ ,

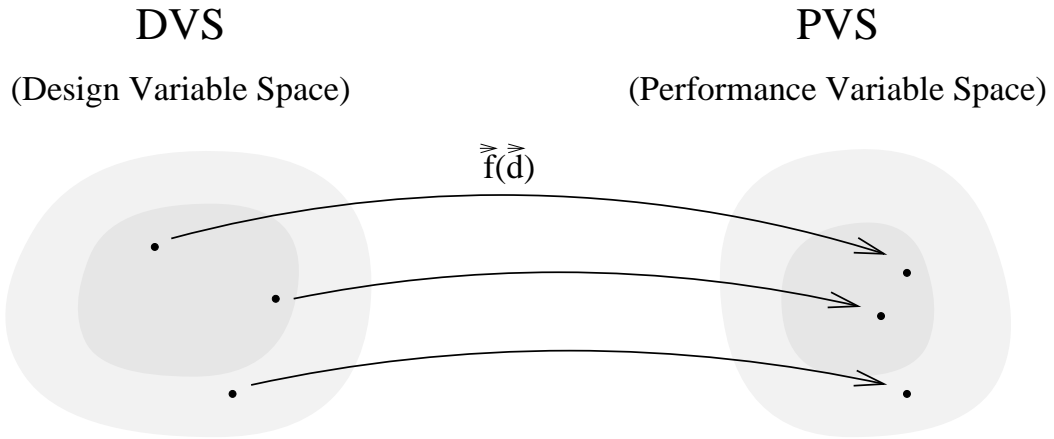
# Chapter 1

## Introduction

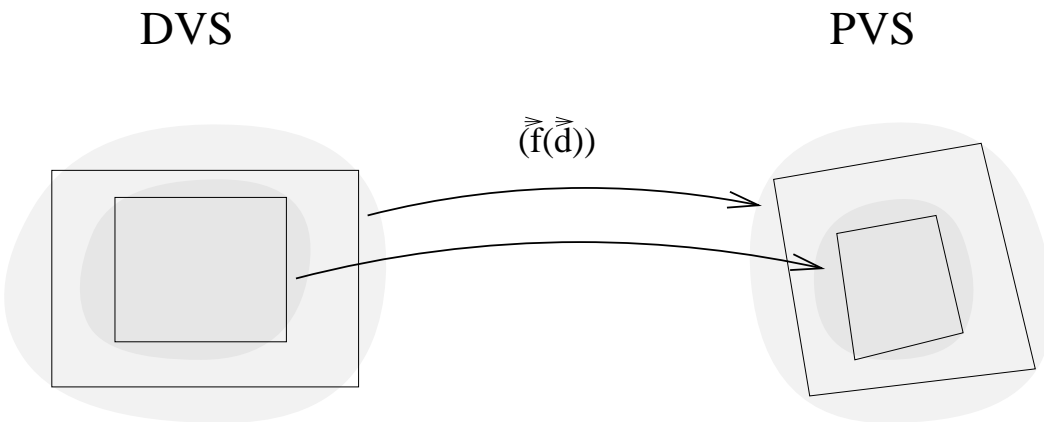
To learn truly what each thing is, is a matter of uncertainty.

*Democritus (ca 460–ca 370 B.C.)*

*Imprecision* is uncertainty that arises because of vague or incomplete information. Preliminary design information is characteristically imprecise: specifications and requirements are subject to change, and the design description is vague and incomplete. Precise information about the final design is usually not available. Yet many powerful evaluation tools, including finite element models, expect precisely specified data. Thus it is common for engineers to evaluate promising designs one by one. Alternatively, optimization may be used to search for the single “best” design. But these approaches focus on individual, precisely specified points in the design space and provide limited information about the full range of possible designs under consideration. This is illustrated in Figure 1.1, where individual designs  $\vec{d}$  in the *design variable space (DVS)* are evaluated through a multi-dimensional mapping  $\vec{f}(\vec{d})$  to obtain multiple measures of design performance in the *performance variable space (PVS)*. (The DVS is not necessarily Euclidean, *i.e.*,  $\mathbb{R}^n$ , though the PVS is usually assumed to be.) The information provided is local to the individual design points evaluated and short of evaluating a large number of points there is no systematic provision for exploring the full space of designs.



**Figure 1.1** Evaluating individual designs one by one.



**Figure 1.2** Evaluating sets of designs.

An alternative approach would be to evaluate *sets* of designs, as illustrated in Figure 1.2. This seeks to provide information on the full range of acceptable designs. By distinguishing subsets of designs that are in various ways preferred, this approach can explicitly model design imprecision.

The *method of imprecision* borrows the notion of membership in a fuzzy set in order to represent preferences among designs. Fuzzy sets model uncertainty in categorization. The set of tall men, for example, is fuzzy in that its boundaries are not precisely defined: it is inaccurate to assume that a single, crisp height, *e.g.*, 5 feet 9 inches, sharply distinguishes tall men from not-tall men. Instead, membership, a real number between zero and one, defines the degree to which an individual belongs to the set. In this example, the fuzziness associated with the set of tall men is linguistic and stems from the inherent fuzziness in the definition of “tall men.” Design imprecision is subtly different. The fuzziness associated with a design specification is not fundamentally linguistic. Imprecise design information is not fuzzy in meaning, but fuzzy in unresolved alternatives. Design imprecision is progressively reduced through design decisions until, ultimately, the final design is precisely specified. Early in the design process, it is not clear to what degree each design alternative reflects the final design that is as yet unknown. Although design imprecision is not a form of uncertainty in meaning, it is still a form of uncertainty in categorization, for which fuzzy sets are an appropriate representation. A direct application of fuzzy set theory would focus on the membership of each design alternative in the set of possible final designs, presumably also the set of best possible designs. This is, however, a somewhat esoteric notion and thus instead of membership, the method of imprecision focuses on preferences: the actual or anticipated preferences of the customer. This is a distinction in interpretation rather than in mathematics, but it is nevertheless significant.

Simon French [17] has questioned the value of fuzzy sets in a *normative* or prescriptive theory of decision-making. In contrast to a descriptive decision analysis, a normative decision analysis seeks to advise or guide the decision-maker. French [17] distinguishes two ways in which a normative analysis can guide decisions:

1. By example — through constructing a *model decision problem*, invoking a *model decision-maker*, and hence arriving at an idealized yet representative decision.
2. Through the modeling process itself — defining a model decision-maker encourages the exploration and clarification of the decision-maker’s own preferences and beliefs.

In this context, French [17] has raised three concerns, posed as questions, for fuzzy decision analysis:

1. Is the model decision problem erected within a fuzzy analysis a suitable representation for real problems?
2. Do I (as the decision-maker) wish to emulate the “ideal” behavior exhibited by the model decision-maker, *i.e.*, do I accept the underlying canons of rationality?
3. Is the process of constructing the model decision-maker as a reflection of me both feasible and informative; and is it helpful in guiding the evolution of my beliefs and preferences?

These three questions will be used as a basis for discussing the contributions of this thesis. The practical value of the methods developed in this work will be demonstrated to directly address French’s concerns.

## 1.1 Organization of Thesis

This thesis builds on the work of Wood and Antonsson [66, 67, 68, 69] and Otto and Antonsson [43, 44, 46, 47, 69]. Their work has laid a broad theoretical foundation for the method of imprecision. The work described in this thesis seeks, through examining the specific rather than the general, and the practical rather than the theoretical, to bring the method closer to implementation in industry. Its two principal contributions are a clearer interpretation of the elements that comprise the method and a practical and efficient computational implementation.

Chapter 2 is concerned with modeling imprecision in engineering design. Sections 2.1 and 2.2 present key definitions that form the foundation for the model decision problem that the method assumes. In particular, the modeling of imprecision in terms of preference is defined. The model decision-maker is introduced in Sections 2.3 to 2.6. Axioms for rational design decision-making are presented in Section 2.3. Section 2.4 discusses importance weighting. In Section 2.7, an electric vehicle design example is presented to illustrate not only the model decision problem and model decision-maker, but also the process by which these models can be constructed. Section 2.8 surveys related work in design decision-making under uncertainty.

Chapter 3 describes the algorithms used to perform preference calculations and discusses important issues in implementing the method of imprecision in a computational tool. Section 3.1 describes previous work and motivates the development of improved methods based on optimization (Section 3.2) and design of experiments (Section 3.4). The particular difficulty addressed, that of mapping preference from the design variables to performance variables, is discussed in detail in Section 3.3.

Chapter 4 introduces the Imprecise Design Tool, a computer program developed by the author that implements the method of imprecision in order to verify the algorithms described in Chapter 3 and to apply the method examples taken from industry. Two industrial examples are presented: the first from aircraft engine development (Section 4.1) and the second from automobile body design (Sections 4.2).

Chapter 5 discusses wider issues involved in implementing the method in industry. The interpretation and specification of preferences is addressed in Section 5.1. Section 5.2 presents a scenario for implementation involving the electric vehicle example from Section 2.7. An informal procedure for supporting group decisions is discussed. Section 5.3 lists the essential steps in the method of imprecision as presented in the electric vehicle design scenario.

Chapter 6 summarizes the contributions of this thesis and returns to French's three concerns for fuzzy decision analysis. The work presented in this thesis addresses, within the limited context of design decision-making under imprecision,

each of French's concerns.

Appendix A describes an algorithm to approximate a mapping through successively subdividing the search space into tiles. This method ultimately proved to have severe limitations. It is included for completeness and as an anecdote that speaks honestly about the reality of research.

## Chapter 2

# Modeling Imprecision in Design

“Where shall I begin, please your Majesty?” he asked.

“Begin at the beginning,” the King said, gravely, “and go on till you come to the end: then stop.”

*Lewis Carroll* (1832–1898), “Alice’s Adventures in Wonderland”

This chapter begins with fundamental definitions that underpin the model decision problem assumed by the method of imprecision. The notion of preference and how it is used to represent imprecision is introduced in Section 2.2. The model decision-maker is manifested as the aggregation functions that trade-off preference. Section 2.3 presents axioms that attempt to define necessary conditions for aggregation functions to exhibit rationality in design decision-making. Section 2.4 discusses importance weighting and re-casts the axioms of rational design decision-making to include weights. The suitability of the model decision problem constructed is supported by examples throughout the chapter. A more detailed example involving the design of an electric vehicle is presented in Section 2.7. Section 2.8 surveys related work in design decision-making under uncertainty by other research groups.

### 2.1 Basic Definitions

**Definition 2.1** The *design variable space* or *DVS* is the set of design alternatives currently under consideration. □

**Definition 2.2** The *design variables*  $d_1, \dots, d_n$  are the attributes that distinguish alternative designs in the DVS.  $\square$

Design variables need not be continuous: the design variable *styling* may have the discrete values “conservative” and “sporty.” The interval methods used to calculate imprecision, however, require that discrete design variables are at least ordinal. Design variables do not completely specify a design: they serve only to distinguish alternatives. Other attributes of the design either are not under active consideration and have fixed values, or cannot be directly specified and have uncontrolled values. Design variables serve to distinguish design alternatives that the designer considers to be distinct for the purpose of analysis. Hence if the designer is considering different lengths of a particular component, then that length should be a design variable. Other variables, such as the width of the same component, do not need to be design variables if they are not under active consideration.

Design variables should be independent: no  $d_i$  should be a function of the other  $d_1, \dots, d_{i-1}, d_{i+1}, \dots, d_n$ . This does not imply that variables cannot be related to each other in any way, but merely that no variable be redundant. For example, two design variables *inner diameter*  $d_1$  and *outer diameter*  $d_2$  are clearly related in that  $d_1 < d_2$ , but neither is a function of the other. *Tube thickness*  $d_3$ , however, is a function of  $d_1$  and  $d_2$  and should not be defined as a third design variable. Note that in this particular example, it would probably be more convenient to choose as design variables *tube thickness* and then either *inner diameter* or *outer diameter*, to avoid having to ensure that *inner diameter* was less than *outer diameter*. The choice of design variables is thus not unique.

The set of valid values for the design variable  $d_i$  is denoted  $\mathcal{X}_i$ . The whole set of design variables forms an  $n$  vector,  $\vec{d}$ , that distinguishes a particular design alternative in the DVS. Distinct  $\vec{d}$  define distinct design alternatives. Conversely, distinct design alternatives will be described by distinct  $\vec{d}$ .

In order to eliminate inferior design alternatives and refine the set of designs under consideration, designs need to be evaluated. Design evaluation seeks to predict how well a design alternative will perform when it is ultimately evaluated by the

customer. “Performance,” in general, has many aspects: rarely is it feasible to quantify all of them.

**Definition 2.3** The *performance variables*  $p_1, \dots, p_q$  are the aspects of a design’s performance that are explicitly quantified. Each performance variable  $p_j$  is defined by a mapping  $f_j$  such that  $p_j = f_j(\vec{d})$ .  $\square$

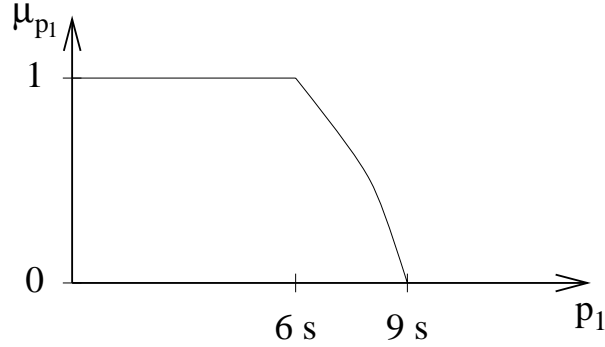
The mappings  $f_1, \dots, f_q$  can be any calculation or procedure to evaluate the performance of a design, including closed-form equations, computational algorithms, “black box” functions, prototype testing, and market research. A design variable can also be a measure of design performance and hence a performance variable. *Weight*, for example, could be a design variable describing distinct design alternatives while also being a performance variable that the customer is interested in. The set of valid values for a performance variable  $p_j$  is denoted  $\mathcal{Y}_j$ . The set of performance variables for each design alternative forms a  $q$  vector,  $\vec{p} = \vec{f}(\vec{d})$ , that specifies the quantified performances of a design  $\vec{d}$ . Other aspects of performance which are not quantified are not formally modeled as performance variables, and are excluded from  $\vec{p}$ .

**Definition 2.4** The *performance variable space* or *PVS* is the set of all quantified performances  $\vec{p} = \vec{f}(\vec{d})$  that are achievable by designs  $\vec{d} \in \text{DVS}$ .  $\square$

## 2.2 Representing Imprecision

Design and performance variables are initially imprecise: they may potentially assume any value within a possible range because the designer does not know, *a priori*, the final value that will emerge from the design process. Yet even though the designer does not know which value will ultimately be specified, certain values will be preferred over others. This preference is used to quantify the imprecision associated with a variable.

Performance variables attempt to predict how the design will perform in the eyes of the customer, and hence for performance variables it is the customer’s preferences



**Figure 2.1** Imprecise functional requirement “less than 8 seconds”.

that should be quantified, even if it is the designer who estimates them. Thus preferences on the time required to accelerate from 0 to 60 mph for an automobile, for example, are the customer’s anticipated preferences, even if the customer has no stated preference on the 0–60 time in particular: the 0–60 time correlates with the average acceleration over the speed range of the engine [37], which in turn correlates with perceived acceleration and vehicle performance. These issues, which together determine preferences on performance variables, will be referred to as *performance considerations*.

**Definition 2.5** The *functional requirement*  $\mu_{p_j}(p_j)$  represents the preference that a customer has for values of the performance variable  $p_j$ :

$$\mu_{p_j}(p_j) : \mathcal{Y}_j \rightarrow [0, 1] \subset \mathbb{R}. \quad \square$$

$\mu_{p_j}(p_j)$  quantifies the customer’s preference for values of  $p_j$  and is distinct from the customary membership function in a fuzzy set, which quantifies the extent to which values belong to the set. An example functional requirement on the performance variable *0–60 time* might at first be given as “less than 8 seconds.” Further elicitation would reveal the imprecision associated with the nominally crisp value “8 seconds,” resulting in a preference function  $\mu_{p_1}$ , where  $p_1$  is *0–60 time*, as shown in Figure 2.1. Values of  $p_1 \leq 6\text{ s}$  have  $\mu_{p_1} = 1$  and are most preferred or ideal. Values

of  $p_1 \geq 9$ s have  $\mu_{p_1} = 0$  and are unacceptable.

The basis for preferences among values of a design variable is less obvious. Design variables distinguish alternative designs: two different values of a design variable  $d_i$  indicate two distinct designs, but the values of  $d_i$  do not indicate if one design is preferred over the other. Design variables do not directly measure design performance and hence the customer has no direct basis for preferring any particular design variable value. The length of the rear axle of an automobile, for example, is not a variable of much interest to the typical customer. Yet choosing different axle lengths will affect aspects of design performance such as vehicle handling that *are* of interest to the customer. Some of these aspects will be quantified and hence modeled as performance variables. The customer's preferences on these performance considerations are already represented as functional requirements. Other aspects of design performance that are not directly related to explicitly quantified performance variables are as yet not modeled and the preferences that exist on these attributes have not been represented. These preferences, corresponding to aspects of design performance that are not explicitly modeled as performance variables, are represented as preferences on the design variables  $d_1, \dots, d_n$ .

**Definition 2.6** The *design preference* function  $\mu_{d_i}(d_i)$  represents the preference that the designer has for values of the design variable  $d_i$  based on aspects of design performance that are not already represented by performance variables:

$$\mu_{d_i}(d_i) : \mathcal{X}_i \rightarrow [0, 1] \subset \mathbb{R}. \quad \square$$

Because the customer has no direct basis for preferences among values of a design variable  $d_i$ , the designer must decide how values of  $d_i$  influence unquantified aspects of design performance which are not represented by performance variables. Specifying a design preference  $\mu_{d_i}$  relies on the designer's experience and judgement in three ways:

1. to determine which unquantified aspects of performance to consider and their relative importance,

2. to estimate how values of the design variable  $d_i$  affect each unquantified aspect of performance considered, and
3. to anticipate the customer's preferences on these unquantified aspects of performance.

In specifying a preference function on the design variable *rear axle length*, the designer might consider that vehicle cornering, suspension geometry constraints, and manufacturability are the most important unquantified aspects of performance. Additionally, experience may lead the designer to define minimum and maximum lengths based on previously successful and unsuccessful vehicle drivetrain designs. These issues, which guide the specification of design preference, will be referred to as *design considerations*. Because vehicle cornering, suspension geometry, manufacturability, and the experience gained from previous vehicle drivetrain designs will not be explicitly quantified, design preferences are the only means of including these important aspects of design performance. Design preferences represent preferences on relevant aspects of design performance that are not explicitly quantified and hence would otherwise be omitted.

### 2.3 Aggregating Preferences

In order to evaluate designs  $\vec{d} \in \text{DVS}$ , the various individual preferences must be combined or aggregated to give a single, overall measure.

**Definition 2.7** The *overall preference*  $\mu_o(\vec{d})$  combines the preferences of the designer and customer for a particular design  $\vec{d}$  and is a function of the design preferences  $\mu_{d_i}(d_i)$ , and the functional requirements  $\mu_{p_j}(p_j) = \mu_{p_j}(f_j(\vec{d}))$ :

$$\mu_o(\vec{d}) = \mathcal{P} \left( \mu_{d_1}(d_1), \dots, \mu_{d_n}(d_n), \mu_{p_1}(f_1(\vec{d})), \dots, \mu_{p_q}(f_q(\vec{d})) \right). \quad \square$$

The *aggregation function*  $\mathcal{P}$  reflects the trade-off strategy, which indicates how competing attributes of the design should be traded-off against each other [44, 45].

An airplane can be made lighter, but this action will probably increase manufacturing cost. One of the most difficult aspects of product development is recognizing, understanding, and managing such trade-offs in a way that maximizes the success of the product. [60] (p5)

The trade-off strategy formalizes the designer's balancing of conflicting goals and constraints. The overall preference embodies the preferences that the designer expresses on design variables as well as the preferences that the customer expresses on the performance variables. Overall preference may be expressed on the DVS,  $\mu_o(\vec{d})$ , or on the PVS,  $\mu_o(\vec{p})$ :  $\mu_o(\vec{p})$  will be defined in Section 3.1. The set of design configurations that maximize  $\mu_o$  is denoted  $\mathcal{X}^*$ . Such peak preference designs  $\vec{d} \in \mathcal{X}^*$  are "most preferred" with respect to the design and performance variables modeled:

$$\forall \vec{d} \in \mathcal{X}^* \quad \mu_o(\vec{d}) = \mu_o^* = \sup\{\mu_o(\vec{d}) \mid \vec{d} \in \text{DVS}\}.$$

$\mu_o^*$  is the peak overall preference in both the DVS and the PVS [47]. The set of performances that correspond to the set of peak preference designs  $\mathcal{X}^*$  is denoted  $\mathcal{Y}^* = \vec{f}(\mathcal{X}^*)$ .

The following five axioms have been suggested as necessary conditions in order for  $\mathcal{P}$  to reflect how a designer might rationally trade-off preferences [43] (set  $N = p + q$ ):

**Axiom 2.8** *Commutativity:*

$$\mathcal{P}(\mu_1, \dots, \mu_j, \dots, \mu_k, \dots, \mu_N) = \mathcal{P}(\mu_1, \dots, \mu_k, \dots, \mu_j, \dots, \mu_N) \quad \forall j, k.$$

A basic condition is that the overall preference should not depend on the order of the preferences being combined.

**Axiom 2.9** *Monotonicity:*

$$\mathcal{P}(\mu_1, \dots, \mu_k, \dots, \mu_N) \leq \mathcal{P}(\mu_1, \dots, \mu'_k, \dots, \mu_N) \quad \text{for } \mu_k \leq \mu'_k \quad \forall k.$$

As any single preference increases or decreases, the overall preference should either move in the same direction, or not at all. An aggregation function that does not satisfy monotonicity would allow a design with the same preferences as a second design but with a lower preference on one particular variable, to have a higher overall preference. If two bicycle designs differ only in that one is lighter (hence higher preference on the performance variable *weight*), then only an irrational trade-off strategy would prefer the heavier design.

**Axiom 2.10** *Continuity:*

$$\mathcal{P}(\mu_1, \dots, \mu_k, \dots, \mu_N) = \lim_{\mu'_k \rightarrow \mu_k} \mathcal{P}(\mu_1, \dots, \mu'_k, \dots, \mu_N) \quad \forall k.$$

Two designs with the same preferences on all variables except one, for which the preferences differ infinitesimally, should have similar overall preferences. An aggregation function should not create discontinuities in the overall preference where there are no discontinuities in the preferences that are being aggregated.

**Axiom 2.11** *Idempotency:*

$$\mathcal{P}(\mu, \dots, \mu) = \mu.$$

If all aspects of a design are equally satisfactory and have the same preference  $\mu$ , then the overall preference should also be  $\mu$ . A non-idempotent aggregation function would be either optimistic or pessimistic in aggregating preferences and would introduce an artificial bias.

**Axiom 2.12** *Annihilation:*

$$\mathcal{P}(\mu_1, \dots, 0, \dots, \mu_N) = 0.$$

A preference of 0 is defined as representing complete dissatisfaction: the variable value specified is unacceptable. Unacceptability implies that the design has failed to meet a minimum requirement. If any aspect of a design is in this way unacceptable,

the entire design must be unacceptable. If, for example, an electric vehicle design has unacceptably high structural stresses, no enhancement in cost, vehicle performance, range, or styling can compensate.

These design axioms do not define rational decision-making in general, but merely indicate necessary conditions for aggregation functions within a fuzzy model of engineering design such that these functions appropriately reflect how designers rationally aggregate preferences. Fung and Fu [18] define a similar set of axioms for rational decision-making in general: commutativity, monotonicity, continuity, idempotency, and associativity. With the exception of associativity, which is included in the definition of hierarchical aggregation functions in Section 2.6, these axioms are a subset of the five design axioms. It is apparent that the annihilation axiom is particular to design. Yet it is a corollary to the definition of zero preference as failure to meet a minimum requirement. The annihilation axiom is necessary to ensure that every acceptable design meets all minimum requirements.

Aggregation functions that satisfy the five design axioms shall be termed *design-appropriate*. A variety of design-appropriate aggregation functions exist. The choice of aggregation function is, however, not one that the designer is free to make: the appropriate trade-off strategy is usually dictated by the design problem. Although it is the designer who balances the different attributes of the design, it is the relationship between attributes, a property of the design problem itself, that determines how they should be traded-off.

Consider a system of components, where the failure of one component results in the failure of the system such that the entire assembly must be replaced. A high preference corresponding to a long time to failure for one component cannot compensate for a low preference corresponding to a short time to failure for another component. Only the lowest preference should be considered in evaluating the design: higher preferences for other attributes of the design cannot compensate for a lower preference. This is a *non-compensating* trade-off strategy for which the

aggregation function is the minimum  $\mathcal{P}_{\min}$ :

$$(2.1) \quad \mu_o(\vec{d}) = \min(\mu_{d_1}, \dots, \mu_{d_n}, \mu_{p_1}, \dots, \mu_{p_q}).$$

This is Bellman and Zadeh's [6] hard "and" operation for fuzzy sets, which does not allow attributes to be traded-off against each other. Yager [70] notes that this choice of aggregation function leads to the classic max-min solution from game theory.

Alternatively, consider an ordinary household battery, and in particular the performance variables battery life (energy stored) and unit cost. A different relationship exists between these two variables. Low unit cost can partially compensate for short battery life and long battery life can partially compensate for high unit cost. Name brand alkaline batteries are examples of long life, high cost designs. Generic batteries are examples of low cost, short battery life designs. The two attributes unit cost and battery life can be traded-off against each other, so that a more acceptable attribute partially compensates for a less acceptable attribute. This can be modeled as a *fully compensating* trade-off strategy for which the aggregation function is the geometric weighted mean or product of powers  $\mathcal{P}_{\Pi}$ :

$$(2.2) \quad \mu_o(\vec{d}) = \left( \prod_{i=1}^n \mu_{d_i} \prod_{j=1}^q \mu_{p_j} \right)^{\frac{1}{n+q}}.$$

This is Bellman and Zadeh's [6] soft "and" operation for fuzzy sets, which corresponds to the Nash solution from game theory [70].

The aggregation functions  $\mathcal{P}_{\min}$  and  $\mathcal{P}_{\Pi}$ , which correspond to non-compensating and fully compensating trade-off strategies, are two limiting cases in a family of design-appropriate aggregation functions identified by Scott and Antonsson [54]. This class of functions will be introduced in Section 2.5.

## 2.4 Weights

The relative importance of different attributes of the design must be considered in combining their corresponding preferences. This is achieved by assigning individual *weights* to each variable:

$$\begin{aligned}\omega_{d_i} &\geq 0 \\ \omega_{p_j} &\geq 0.\end{aligned}$$

Each weight  $\omega$  quantifies the importance of its associated variable *relative* to other variables.

It has been proposed that importance is a function of design and performance variables [43]. Consider, for example, one link of a multi-link vehicle suspension system. The longitudinal stress in this component becomes an important, indeed critical, variable as it approaches the yield stress of the material. Yet ordinarily the stress in this particular component would not be considered especially important. This suggests that the corresponding importance weighting should vary with the stress in order to correctly represent the change in perceived importance. But this is not necessary within the method of imprecision because specifying a preference of zero or close to zero ensures that the stress in this component becomes the critical attribute, because of the axioms of annihilation and continuity. The variation of importance with a variable's value reflects the shifting criticality of variables relative to each other. This shift in criticality is already modeled by preference functions on design and performance variables. Weights need only model the relative importance of variables within the context of the design problem, without reference to specific designs  $\vec{d}$  or performances  $\vec{p}$ .

In order to account for relative importance, aggregation functions must now aggregate preference and weight, *i.e.*,  $(\mu, \omega)$  pairs. Note that preferences are functions of the variables they represent, but weights are constants. The axioms for design-appropriateness must be redefined to include weights:

**Axiom 2.13** *Commutativity:*

$$\begin{aligned} \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_j, \omega_j), \dots, (\mu_k, \omega_k), \dots, (\mu_N, \omega_N)) = \\ \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, \omega_k), \dots, (\mu_j, \omega_j), \dots, (\mu_N, \omega_N)) \quad \forall j, k. \end{aligned}$$

**Axiom 2.14** *Monotonicity:*

$$\begin{aligned} \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, \omega_k), \dots, (\mu_N, \omega_N)) \leq \mathcal{P}((\mu_1, \omega_1), \dots, (\mu'_k, \omega_k), \dots, (\mu_N, \omega_N)) \\ \text{for } \mu_k < \mu'_k \quad \forall k \\ \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, \omega_k), \dots, (\mu_N, \omega_N)) \leq \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, \omega'_k), \dots, (\mu_N, \omega_N)) \\ \text{for } \omega_k < \omega'_k \text{ where } \mu_j \leq \mu_k \quad \forall j \neq k \quad \forall k. \end{aligned}$$

Increasing the importance of the most preferred attribute should not decrease the overall preference.

**Axiom 2.15** *Continuity:*

$$\begin{aligned} \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, \omega_k), \dots, (\mu_N, \omega_N)) = \\ \lim_{\mu'_k \rightarrow \mu_k} \mathcal{P}((\mu_1, \omega_1), \dots, (\mu'_k, \omega_k), \dots, (\mu_N, \omega_N)) \quad \forall k \\ \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, \omega_k), \dots, (\mu_N, \omega_N)) = \\ \lim_{\omega'_k \rightarrow \omega_k} \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, \omega'_k), \dots, (\mu_N, \omega_N)) \quad \forall k. \end{aligned}$$

Aggregation functions should be continuous in preferences and in weights.

**Axiom 2.16** *Idempotency:*

$$\mathcal{P}((\mu, \omega_1), \dots, (\mu, \omega_N)) = \mu.$$

**Axiom 2.17** *Annihilation:*

$$\begin{aligned} \mathcal{P}((\mu_1, \omega_1), \dots, (0, \omega_k), \dots, (\mu_N, \omega_N)) &= 0 \quad \text{where } \omega_k \neq 0 \quad \forall k \\ \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_k, 0), \dots, (\mu_N, \omega_N)) &= \\ \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_{k-1}, \omega_{k-1}), (\mu_{k+1}, \omega_{k+1}), \dots, (\mu_N, \omega_N)) &\quad \forall k. \end{aligned}$$

A weight of 0 is defined as removing the attribute from consideration.

Weights have been defined without an upper bound on their value and without the necessity for normalization. Thus an additional axiom is required to correctly aggregate weights.

**Axiom 2.18** *Self-normalization:*

$$\mathcal{P}((\mu_1, \lambda\omega_1), \dots, (\mu_N, \lambda\omega_N)) = \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) \quad \text{where } \lambda > 0.$$

Self-normalization allows groups of weights to be freely scaled by any strictly positive constant  $\lambda$ . This is a necessary property for hierarchical aggregation, which is discussed in Section 2.6. Design-appropriate weighted aggregation functions must satisfy Axioms 2.13–2.18. Note that non-weighted aggregation functions are a special case of weighted aggregation functions with uniform weights.

## 2.5 Weighted Means

This section discusses the class of functions known as the (quasilinear) weighted means [1]:

$$(2.3) \quad \mathcal{P}((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = g \left( \frac{\omega_1 g^{-1}(\mu_1) + \dots + \omega_N g^{-1}(\mu_N)}{\omega_1 + \dots + \omega_N} \right)$$

where  $g$  is a strictly monotonic, continuous generating function with inverse  $g^{-1}$ ;  $g(0) \leq \mu_1, \dots, \mu_N \leq g(1)$ ;  $\omega_1, \dots, \omega_N \geq 0$ ; and  $\omega_1 + \dots + \omega_N > 0$ . Scott and Antonsson [54] show that the properties of the weighted mean include all the properties of design-appropriate weighted aggregation functions except for annihilation (Ax-

iom 2.17). Thus any weighted mean that satisfies annihilation is design-appropriate. Properties of the weighted mean that are sufficient to define the form of Equation (2.3) can be derived from the remaining design axioms (2.13–2.16, and 2.18) with the additional assumption of strict monotonicity [54]. Hence any strictly monotonic design-appropriate aggregation function must be a weighted mean.

The weighted root-mean-power family of functions is generated by the function  $g(\mu) = \mu^s$  [1]:

$$(2.4) \quad \mathcal{P}_s((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = \left( \frac{\omega_1 \mu_1^s + \dots + \omega_N \mu_N^s}{\omega_1 + \dots + \omega_N} \right)^{\frac{1}{s}}$$

where the parameter  $s \in \mathbb{R}$  and  $g(0) = 0 \leq \mu_1, \dots, \mu_N \leq g(1) = 1$ . Weighted root-mean-power functions that satisfy annihilation are design-appropriate. A weighted mean satisfies annihilation if and only if  $\lim_{\mu \rightarrow 0} g^{-1}(\mu)$  is unbounded [54].  $\lim_{\mu \rightarrow 0} \mu^{\frac{1}{s}}$  is unbounded for  $s \leq 0$  only. Hence  $\mathcal{P}_s$  where  $s \leq 0$  is a class of design-appropriate aggregation functions.

Consider the limiting cases of  $s = 0$  and  $s = -\infty$ .  $\mathcal{P}_{s=0}$  is the product of powers  $\mathcal{P}_{\Pi}$  in its weighted form [54]:

$$(2.5) \quad \mathcal{P}_{s=0}((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = \left( \prod_{k=1}^N \mu_k^{\omega_k} \right)^{\frac{1}{\omega}} \quad \text{where } \omega = \omega_1 + \dots + \omega_N.$$

$\mathcal{P}_{s=-\infty}$  is  $\mathcal{P}_{\min}$  without weights [54]:

$$(2.6) \quad \mathcal{P}_{s=-\infty}((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = \min(\mu_1, \dots, \mu_N).$$

$\mathcal{P}_{s \leq 0}$  interpolates between the non-compensating and fully compensating trade-off. The degree of compensation increases as  $s$  increases from  $-\infty$ . Intermediate trade-offs corresponding to intermediate values of  $c$  shall be termed *partially compensating*. The class of functions  $\mathcal{P}_{s \leq 0}$  is not unique in interpolating between  $\mathcal{P}_{\min}$  and  $\mathcal{P}_{\Pi}$ : there exist other generating functions that give rise to design-appropriate aggregation functions that trade-off preferences differently [54].

That  $\mathcal{P}_{\min}$  and  $\mathcal{P}_{\Pi}$  define the limits of the family of design-appropriate func-

tions  $\mathcal{P}_{s \leq 0}$  suggests that the non-compensating and fully compensating strategies represent extremes in design-appropriate trade-offs. Indeed, Yager [70] suggests that “these forms may represent in the oriental sense the Yin and the Yang . . . .” Idempotency and monotonicity ensure that no design-appropriate aggregation function can generate values less than  $\mathcal{P}_{\min}$  for any set of input preferences. Thus  $\mathcal{P}_{\min}$  defines a lower bound for design-appropriate functions in general [54]. Idempotency and monotonicity also ensure that the maximum is an upper bound for design-appropriate aggregation functions. The maximum, however, fails to satisfy annihilation and is not design-appropriate. Moreover, a function  $\max'$  defined as equal to  $\max$  except where annihilation requires  $\max' = 0$ , would fail to satisfy continuity where the function transitions [54]. Thus a maximal design-appropriate aggregation function  $\mathcal{P}_{\max'}$  would be as close as possible to  $\max'$  while maintaining continuity near  $\mu_k = 0$ .  $\mathcal{P}_{\Pi}$  does not define an upper bound for design-appropriate aggregation functions, even though it defines the upper bound of a particular class of design-appropriate aggregation functions.

The class of functions  $\mathcal{P}_{s > 0}$  do not satisfy annihilation, but are compatible with the remaining design axioms. Therefore, a class of design-appropriate aggregation functions could be loosely defined in a similar manner as the maximal design-appropriate aggregation function described above:

$$(2.7) \quad \mathcal{P}_{s > 0'}((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = \begin{cases} \mathcal{P}_{s > 0} & \text{if } \mu_1, \dots, \mu_N \geq \delta \\ 0 & \text{if } \mu_k = 0, \omega_k \neq 0, 1 \leq k \leq N \\ \mathcal{P}_{\delta} & \text{otherwise} \end{cases}$$

where  $0 < \delta \ll 1$  and  $\mathcal{P}_{\delta}$  continuously interpolates between  $\mathcal{P}_{s > 0'} = 0$  at  $\mu_k = 0$  and  $\mathcal{P}_{s > 0'} = \mathcal{P}_{s > 0}$  at  $\mu_k = \delta$ . In practice, it is not necessary to define  $\delta$  or  $\mathcal{P}_{\delta}$  except that  $\delta$  is distinguishably greater than 0, but is less than the lowest distinguishably greater than 0 preference specified (*i.e.*,  $\delta$  is small but not infinitesimal). The discretization of preferences that obviates a precise definition of  $\mathcal{P}_{s > 0'}$  is discussed in Section 5.1.

$\mathcal{P}_{s > 0'}$  interpolates between  $\mathcal{P}_{\Pi}$ , the fully compensating trade-off, and  $\mathcal{P}_{\max'}$ , the maximal design-appropriate aggregation function. As  $s$  increases to  $+\infty$  the

degree of compensation increases such that smaller increases in a higher preference compensate for larger decreases in a lower preference. This willingness to trade a small gain for a large loss implies that  $\mathcal{P}_{s>0'}$  is a family of *supercompensating* aggregation functions. At the extreme,  $\mathcal{P}_{\max'}$  judges a design by its best attribute, except where another attribute is close to unacceptable ( $\mu_k < \delta$ ). This may not be irrational, but it is difficult to envision a design problem for which  $\mathcal{P}_{\max'}$  would be an appropriate trade-off strategy.

The parameter  $s$  is unwieldy and not readily interpreted. A parameter  $c$  can be suitably defined to represent the degree of compensation on the interval  $[0, 2]$ :

$$(2.8) \quad c = 1 + \frac{s}{k + |s|} \quad \text{where } k > 0$$

such that

$$(2.9) \quad s(c) = k \frac{c-1}{1-|c-1|} \quad s(0) = -\infty, s(1) = 0, \text{ and } s(2) = +\infty.$$

The constant  $k$  should be set to some positive value such that intermediate values of  $c$  represent appropriate degrees of compensation. A suitable value for  $k$  has not, as yet, been determined. It is future research. The value for  $k$  should be determined in conjunction with a, perhaps informal, definition of what the degree of compensation  $c$  represents. Currently only three values are pinned down:  $c = 0$  is non-compensating,  $c = 1$  is fully compensating, and  $c = 2$  is maximally (super-) compensating. A family of design-appropriate aggregation functions may be defined by combining  $\mathcal{P}_{s \leq 0}$  and  $\mathcal{P}_{s > 0'}$ , and reparameterizing in  $c$ :

$$(2.10) \quad \mathcal{P}_c((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = \begin{cases} \mathcal{P}_{s \leq 0} \quad s = k(1 - \frac{1}{c}) & \text{if } 0 \leq c \leq 1 \\ \mathcal{P}_{s > 0'} \quad s = k(\frac{1}{2-c} - 1) & \text{if } 1 < c \leq 2 \end{cases}.$$

Table 2.1 summarizes the continuum of aggregation functions defined by  $\mathcal{P}_c$ .

non-compensating	$\mathcal{P}_{\min}$	$s = -\infty$	$c = 0$
partially compensating	$\mathcal{P}_{s<0}$	$-\infty < s < 0$	$0 < c < 1$
fully compensating	$\mathcal{P}_{\Pi}$	$s = 0$	$c = 1$
supercompensating	$\mathcal{P}_{s>0'}$	$0 < s < \infty$	$1 < c < 2$
maximally compensating	$\mathcal{P}_{\max'}$	$s = \infty$	$c = 2$

**Table 2.1** Design-appropriate aggregation functions based on the weighted means, parameterized in  $s$  and  $c$ .

## 2.6 Hierarchical Weighted Design

Within a single design problem, different groups of attributes may require different trade-off strategies. In the design of a consumer product, for example, certain variables related to safety might require a non-compensating trade-off, while other variables related to convenience or portability would require a compensating trade-off. In general, preferences for individual attributes will need to be successively aggregated by a hierarchy of different trade-off strategies. Each aggregation operation must aggregate not only preferences but also weights, such that aggregating the (preference, weight) pairs corresponding to individual attributes results in an aggregated (preference, weight) pair. A hat will be used to denote (preference, weight) pairs:  $\hat{\mu} = (\mu, \omega)$ . Aggregation operations that aggregate both preference and weights will also be denoted by a hat:

$$(2.11) \quad \hat{\mathcal{P}}(\hat{\mu}_1, \dots, \hat{\mu}_N) = (\mathcal{P}^\mu(\hat{\mu}_1, \dots, \hat{\mu}_N), \mathcal{P}^\omega(\omega_1, \dots, \omega_N)).$$

Suppose that the  $N = n + q$  design and performance variables are split into two subsets so that a different trade-off strategy can be applied to each:

$$(2.12) \quad \begin{aligned} \hat{\mu}_o &= (\mu_o, \omega_o) \\ &= \hat{\mathcal{P}}(\hat{\mu}_1, \dots, \hat{\mu}_N) \\ &= \hat{\mathcal{P}}_{III} \left( \hat{\mathcal{P}}_I(\hat{\mu}_1, \dots, \hat{\mu}_k), \hat{\mathcal{P}}_{II}(\hat{\mu}_{k+1}, \dots, \hat{\mu}_N) \right). \end{aligned}$$

This is a hierarchical form of Definition 2.7. How should the subordinate aggregation operations  $\widehat{\mathcal{P}}_I$  and  $\widehat{\mathcal{P}}_{II}$  and the superordinate aggregation operation  $\widehat{\mathcal{P}}_{III}$  be defined? If a particular trade-off,  $\mathcal{P}_c$  (Equation (2.10)) with an appropriate degree of compensation  $c$ , is used to aggregate preferences for all  $N$  variables, then  $\mathcal{P}_I^\mu$ ,  $\mathcal{P}_{II}^\mu$ , and  $\mathcal{P}_{III}^\mu$  must satisfy Equation (2.12) for  $\mathcal{P}^\mu = \mathcal{P}_c$ . Yet this condition does not uniquely specify  $\mathcal{P}_I^\mu$ ,  $\mathcal{P}_{II}^\mu$ , and  $\mathcal{P}_{III}^\mu$ , nor does it indicate how weights should be aggregated.

The form of the weighted mean (Equation (2.3)) suggests that aggregated weights should be added if a weighted mean is used to aggregate preferences:

$$\mathcal{P}^\omega(\omega_1, \dots, \omega_N) = \omega_1 + \dots + \omega_N.$$

This definition of  $\mathcal{P}^\omega$  is consistent with Equation (2.12) if  $\mathcal{P}^\mu$ ,  $\mathcal{P}_I^\mu$ ,  $\mathcal{P}_{II}^\mu$ , and  $\mathcal{P}_{III}^\mu$ , are all defined to be the same weighted mean:

$$\begin{aligned} \mathcal{P}^\omega(\mathcal{P}^\omega(\omega_1, \dots, \omega_k), \mathcal{P}^\omega(\omega_{k+1}, \dots, \omega_N)) &= \mathcal{P}^\omega(\omega_1 + \dots + \omega_k, \omega_{k+1} + \dots + \omega_N) \\ &= \omega_1 + \dots + \omega_k + \omega_{k+1} + \dots + \omega_N \\ (2.13) \qquad \qquad \qquad &= \mathcal{P}^\omega(\omega_1 + \dots + \omega_N) \end{aligned}$$

$$\begin{aligned} \mathcal{P}^\mu(\widehat{\mathcal{P}}(\hat{\mu}_1, \dots, \hat{\mu}_k), \widehat{\mathcal{P}}(\hat{\mu}_{k+1}, \dots, \hat{\mu}_N)) &= \mathcal{P}^\mu((\mu_I, \omega_I), (\mu_{II}, \omega_{II})) \\ &= g\left(\frac{\omega_I g^{-1}(\mu_I) + \omega_{II} g^{-1}(\mu_{II})}{\omega_I + \omega_{II}}\right) \\ &= g\left(\frac{(\omega_1 + \dots + \omega_k) g^{-1}\left(g\left(\frac{\omega_1 g^{-1}(\mu_1) + \dots + \omega_k g^{-1}(\mu_k)}{\omega_1 + \dots + \omega_k}\right)\right) + \omega_{II} g^{-1}(\mu_{II})}{\omega_1 + \dots + \omega_k + \omega_{II}}\right) \\ &= g\left(\frac{\omega_1 g^{-1}(\mu_1) + \dots + \omega_k g^{-1}(\mu_k) + \omega_{k+1} g^{-1}(\mu_{k+1}) + \dots + \omega_N g^{-1}(\mu_N)}{\omega_1 + \dots + \omega_k + \omega_{k+1} + \dots + \omega_N}\right) \\ (2.14) \qquad \qquad \qquad &= \mathcal{P}^\mu(\hat{\mu}_1, \dots, \hat{\mu}_N) \end{aligned}$$

Thus the use of an aggregation operation  $(\mathcal{P}^\mu, \mathcal{P}^\omega)$ , where  $\mathcal{P}^\mu$  is a weighted mean and  $\mathcal{P}^\omega$  is the arithmetic sum, has been shown to be hierarchically consistent. The  $\mathcal{P}_s$  family of weighted means generated by  $g(\mu) = \mu^s$  therefore defines hierarchically

consistent aggregation operations, as do the  $\mathcal{P}_{s \leq 0}$  subset of design-appropriate aggregation functions. The specially defined  $\mathcal{P}_{s > 0'}$  (or  $\mathcal{P}_{c > 1}$ ) family of design-appropriate aggregation functions must be separately shown to be hierarchically consistent where  $\mathcal{P}_{s > 0'}$  differs from  $\mathcal{P}_{s > 0}$ , *i.e.*, where there exists a preference  $\mu_k < \delta$ .

Where  $\mu_k = 0$  for some  $k$ , annihilation requires that the aggregated preference be zero and that this zero preference be propagated up the hierarchy. This is clearly satisfied since  $\mathcal{P}_{s > 0'}$  is defined to annihilate such that  $\mathcal{P}_{s > 0'}((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = 0$  if  $\mu_k = 0$  for some  $k$  (Equation (2.7)). Hierarchical consistency need not be shown for the intermediate case where  $0 < \mu_k \leq \delta$  for some  $k$  because such intermediate values of preference do not occur in practice, as is discussed in Section 5.1. Thus  $\widehat{\mathcal{P}}_c$  defined as  $(\mathcal{P}_c, \mathcal{P}^\omega)$ , *i.e.*,  $(\mathcal{P}_{s \leq 0}, \mathcal{P}^\omega)$  and  $(\mathcal{P}_{s > 0'}, \mathcal{P}^\omega)$ , has been shown to be a valid hierarchical weighted aggregation operation for (preference, weight) pairs.

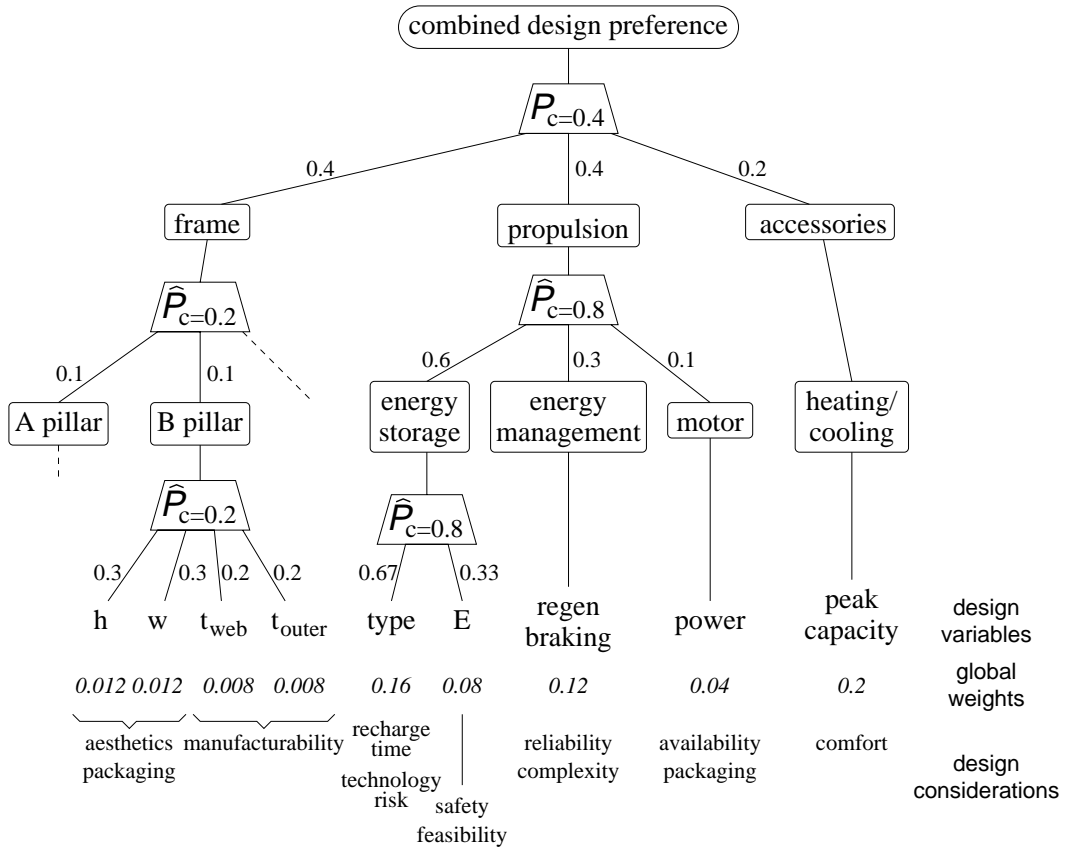
The hierarchical aggregation operation  $\widehat{\mathcal{P}}_c$  can be successively applied, allowing multiple levels of problem decomposition or aggregation. Self-normalization (Axiom 2.18) allows weights to be specified at arbitrary levels of aggregation and freely scaled. A set of variables representing a single design problem can be decomposed into successively smaller sub-problems based on the structure of the problem and the nature of the trade-offs between variables. Importance weightings for each elemental attribute at the lowest level of the hierarchy can be specified either relative to all other elemental attributes globally or relative to attributes within each sub-problem locally. Where weights are specified locally by sub-problem, the aggregated weight representing the importance of the sub-problem must be scaled relative to other sub-problems at the same level of aggregation. This is equivalent to aggregating multiple design problems into a single super-problem: the importance of each problem must be determined relative to the other problems at the same hierarchical level, prior to aggregation. Weights need not be bounded by any specific limit, though it may be convenient to normalize weights within a particular problem to sum to 1, such that they represent the importance of each attribute in the context of the problem as well as relative to each other.



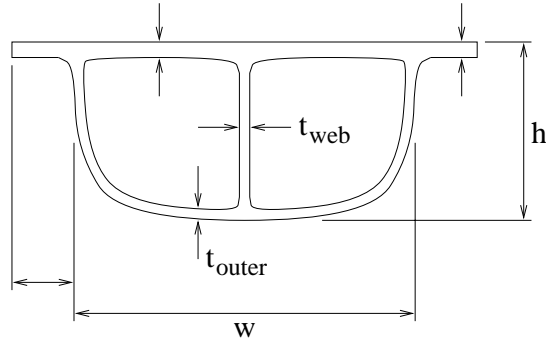
**Figure 2.2** Amerigon's electric vehicle chassis ©*Copyright 1996 CAL-START, Inc., all rights reserved. Permission to use granted April 30, 1996.*

## 2.7 Example: Design of an Electric Vehicle

This section demonstrates the modeling of a hierarchical design problem through an example: the design of an electric vehicle (EV) based on a space frame. It is assumed that the basic geometry of the frame has already been decided and that the design team is currently concerned with basic frame parameters and important choices in vehicle components such as the propulsion system. Figure 2.2 shows a running chassis for an electric vehicle developed by Amerigon Incorporated under the CALSTART business incubator program. This actual working vehicle design is the basis for the example presented here, though the vehicle used for the example is not intended to accurately represent Amerigon's running chassis. The purpose of the running chassis is to provide "a modular 'common platform,' or shared chassis, that can serve as the basis for a family of electric vehicle models for several manufacturers.



**Figure 2.3** Design preference aggregation hierarchy for an electric vehicle design.



**Figure 2.4** B pillar design variables.

The running chassis is a fully functional EV without body or interior, utilizing an aluminum space frame design for lighter weight and lower cost. [23]”

The hierarchical aggregation of individual design preferences is shown in Figure 2.3. Individual design preferences  $\mu_{d_1}(d_1), \dots, \mu_{d_n}(d_n)$  are aggregated through a hierarchy of weighted aggregation functions into the combined design preference  $\mu_d(\vec{d})$  ( $\mu_d(\vec{d})$  will be formally defined in Chapter 3). The A pillars are on either side of the front windscreen, and the B pillars separate the front and rear doors (Figure 2.2). The hierarchy for the frame of the vehicle is incomplete: only the design variables for the B pillar are given. These design variables are shown in Figure 2.4. Note that  $h$ ,  $w$ ,  $t_{web}$ , and  $t_{outer}$  do not fully describe the cross-section. Other variables that are not under active consideration (*e.g.*, those indicated but not labelled) do not need to be modelled as design variables.

Recall that design preferences  $\mu_d(d_i)$  represent the customer’s anticipated preferences with respect to design considerations: the unquantified aspects of design performance which are not represented by performance variables. Thus the first step is to determine which aspects of performance are to be quantified as performance variables. For this example the following performance variables are to be calculated:

- $p_1$  vehicle range

- $p_2$  vehicle cost
- $p_3$  acceleration time from 0–60 mph
- $p_4$  vehicle weight
- $p_5$  frame stiffness in bending
- $p_6$  frame stiffness in torsion

It is assumed that estimates for these quantities are available for any given design configuration. *Range*  $p_1$  can be calculated given data on energy stored, motor and transmission efficiency, aerodynamic drag, and rolling resistance. *Cost*  $p_2$  and *weight*  $p_4$  and can be calculated using a spreadsheet or similar software, given the necessary weight and cost data. *0–60 time*  $p_3$  can be estimated from motor and transmission characteristics. *Bending stiffness*  $p_5$  and *torsional stiffness*  $p_6$  can be evaluated using a finite element model (Section 4.2).

The design considerations that pertain to each design variable are indicated in Figure 2.3. The careful distinction between performance considerations (represented by functional requirements on performance variables) and design considerations (represented by design preferences on design variables) is an important contribution of this thesis. Design considerations explicitly account for relevant aspects of design performance that are otherwise not accounted for as performance considerations. Issues such as aesthetics and manufacturability are clearly relevant to design evaluations, yet they are not easily quantified. Design preferences embody these issues. In constructing the hierarchy to aggregate design preference, it is important to understand which specific issues are being aggregated at each step.

Although the B pillar design variables  $h$  and  $w$  affect stiffness, cost, weight, range, and acceleration, these considerations will ultimately be explicitly evaluated. They do not need to be represented by design preference. The design considerations that remain are aesthetics and packaging. B pillars that are too wide or too narrow are not attractive. B pillars that are too deep reduce clearance inside the vehicle, yet a certain minimum depth and width is required to attach the seat belt. Additionally,

wide B pillars reduce the size of the door opening. The thicknesses  $t_{\text{web}}$  and  $t_{\text{outer}}$  affect the difficulty of extruding the cross-section. The internal web is necessary to maintain the shape of the cross-section when it is bent, although a thick web is difficult to manufacture [11]. The values of  $t_{\text{web}}$  and  $t_{\text{outer}}$  affect the difficulty of extruding a uniform cross-section within tolerance [11]. Joints and attachments to the B pillar lead to a minimum value for  $t_{\text{outer}}$ .

The design considerations associated with the B pillar design variables are not naturally compensating. Packaging constraints and aesthetics cannot significantly compensate for manufacturability. Yet the trade-off is not purely non-compensating. Thus the level of compensation specified is  $c = 0.2$ . The value of  $c$  obtained in this manner is, at best, an estimate. Indeed, the exact parameterization of  $\widehat{\mathcal{P}}_c$  has not yet been determined. Nevertheless, the value  $c = 0.2$  approximately represents the informally defined degree of compensation that is appropriate for the aggregation of the design considerations associated with the B pillar design variables. More precise methods of determining  $c$  will be introduced in Section 5.2.

For each aggregation operation at a particular hierarchical level in Figure 2.3, the locally normalized weights assigned to the aggregated branches are indicated. Global weights normalized for the entire design problem are printed in italics below each design variable. The global weights were obtained by multiplying the local weights along the branches connecting each design variable to the top of the hierarchy. Because it is natural to compare the relative importance of closely related attributes, local weights were specified first. The global weights corresponding to the local weights specified were then examined and adjustments were made to both global and local weights in order to better represent the perceived relative importance of the design variables.

The types of energy storage considered are limited to conventional nickel-cadmium and lead-acid batteries, and an advanced lead-acid battery that uses a lead wire grid extruded onto a fiberglass core. The lead wire-acid battery, developed by Electrosource Incorporated, of Austin, Texas, has advantages in weight and durability and with an estimated price of \$3,000 per vehicle it is among the least expensive new

energy storage options available for electric vehicles [42]. However, there is a degree of risk involved in using a new battery technology that is as yet not in widespread usage. A more important design consideration is the time required to recharge the batteries. Conventional nickel-cadmium batteries can be partially recharged relatively quickly in comparison to conventional lead-acid batteries. The Peugeot 106 and Citroën AX, both produced by PSA Peugeot Citroën, use nickel-cadmium batteries that can be fast-charged in 10 minutes to extend the vehicle's range by 20 to 30 kilometers [42].

The total energy stored,  $E$ , is essentially the number of batteries. As the number of batteries becomes large, the quantity of potentially dangerous acid in a lead-acid battery, for example, becomes a significant safety concern. Battery mass also affects safety in a collision. Additionally, there is an upper limit to the number of batteries that can be physically packed into an electric vehicle, independent of their weight.

A crucial component of a practical electric vehicle is an energy management system. The total energy stored in an electric vehicle is limited: range is dependent on how efficiently that energy is used. Where lead-acid batteries are the only means of energy storage, the maximum power that can be drawn from the batteries is also a critical limitation. An especially important choice is whether or not to use regenerative braking in order to recover some of the energy otherwise dissipated in decelerating the vehicle. Assuming that weight, range, and cost are already accounted for as performance variables, the remaining considerations are the additional complexity of such a system, and in particular its impact on reliability.

The design considerations governing the choice of motor used to propel the vehicle are the availability of types and sizes of motor and packaging constraints that limit the overall dimensions of the motor. These are relatively unimportant issues as is indicated by a global importance of 0.04. Note that this does not imply that the choice of motor is a relatively unimportant decision, nor that the motor is a relatively unimportant component in the vehicle, but merely that availability and packaging constraints for the motor are relatively unimportant compared to other design considerations.

Ironically, the efficiency of an electric vehicle creates a problem that does not exist for combustion-powered vehicles: how to heat the interior when there is no convenient (and free) source of waste heat. The problem is exacerbated by limited energy storage. Thus the passenger heating and cooling system in an electric vehicle is a significant and integral part of the design. The selection of the capacity of the climate control system directly impacts range, cost, and comfort. Of these, comfort is not modeled as a performance variable. Its direct importance to the customer is reflected in a global weight of 0.2. The corresponding design preference  $\mu_{peak\ capacity}$  is more heavily weighted ( $\omega_{peak\ capacity} = 0.2$ ) than any other individual design preference: the design consideration *comfort* associated with *peak capacity* is the most important of the design considerations in Figure 2.3.

Critical aspects of design performance are typically quantified. Thus they are modeled as performance variables and the customer's preferences on them are represented as functional requirements. Therefore, the remaining aspects of design performance that must be expressed as design preferences on design variables are typically of lesser importance. Many design considerations, *e.g.*, manufacturability, are only of indirect interest to the customer. Thus the extent to which the determination of the design preference hierarchy, the relative weights, and the aggregation functions is informal and approximate, is entirely appropriate. The degree of compensation  $c$  for each aggregation operation need only be determined to the nearest 0.1, at most: for many problems the nearest 0.2 will suffice. Perhaps a more valuable result of constructing the design preference hierarchy is the understanding gained through identifying design considerations, their relative importance, and their associated design variables, and formalizing the hierarchical relationships between design considerations. A similar benefit can be expected from constructing the functional requirement hierarchy. The functional requirement hierarchy is more easily constructed because performance considerations are relatively specific and well-defined. It has been demonstrated that the construction of these models is not only feasible and informative, but in necessitating the careful identification of the specific considerations corresponding to each variable and their relative importance,

the process of constructing the model is itself helpful in understanding the structure of preferences that characterizes the design problem.

## 2.8 Related Work

Decision-making methods that address uncertainty, in a broad sense, are not new. The method of imprecision presented in this thesis may be distinguished from these other methods in three principal directions:

- the type of uncertainty modeled,
- the means by which uncertainty is modeled,
- the functions used to aggregate uncertainty.

### Probability

Probability theory quantifies uncertainty due to random variation. Probability methods focus on observing a process in order to characterize its behavior and hence predict the likelihood of various outcomes. Probabilistic uncertainty arises from a stochastic process for which the best predictor of the final outcome is statistical analysis of previous behavior. This is in contrast to imprecision in design, which arises because the designer has yet to make decisions that will more precisely determine the design. This decision-making process is clearly not random and analysis of previous behavior is of limited value. Thus probability methods are not well-suited to modeling design imprecision. Probability methods are, however, well-suited to dealing with stochastic uncertainty in manufacturing processes, material properties, loading, reliability, *etc.* Probabilistic design [22, 56] seeks to support design decisions through the analysis of these stochastic variations.

Taguchi's method [7, 50, 57], which is widely used in industry, views manufacturing variations as undesirable stochastic noise. The method has three key aspects [50]:

1. The loss in producing a product that deviates from target values is assumed to be quadratic.
2. Selected experiment design techniques are used to efficiently characterize the behavior of the manufacturing process relative to controlled inputs.
3. The goal is to achieve robustness both in terms of the insensitivity of product performance to uncontrolled variation as well as the consistency of the manufacturing process in delivering products to specification.

Taguchi’s method is a philosophy to understand and minimize the cost of stochastic process variations and, as such, has been shown to be effective. Design imprecision, however, is not explicitly modeled in the Taguchi approach.

### **Utility Theory**

Utility theory seeks to aid decision-making in the presence of uncertainty. The type of uncertainty modeled is uncertainty due to decisions yet to be made, *i.e.*, imprecision. Utility theory is based on economics and its central assumption is that each aspect of a decision can be assigned a function representing utility. Although utility is similar to preference as used in the method of imprecision (von Neumann and Morgenstern [61] use “satisfaction” and “preference” as similes for utility), there are three important differences:

1. Utility functions are specified only on objectives: where there are multiple courses of action, for example, the expected utility of each action is assessed on each objective variable [25]. The method of imprecision admits a second possibility: that preferences may also be specified on design variables, based on anticipated design performance. Proxy attributes in utility theory [25] are equivalent to performance variables and not design variables because they still relate directly to objectives, and most significantly the mapping from proxy attributes to objectives is not explicitly evaluated.
2. Utility is based on a common monetary commodity:

We shall therefore assume that the aim of all participants in the economic system, consumers as well as entrepreneurs, is money, or equivalently a single monetary commodity. This is supposedly to be unrestrictedly divisible and substitutable, freely transferable and identical, even in the quantitative sense, with whatever “satisfaction” or “utility” is desired by each participant. [61] (p8)

While this is reasonable in the context of economic systems, it is not especially suited to design. Preferences on design attributes are not necessarily identical and substitutable commodities. Preferences on the stresses in various components of an automobile suspension system are not identical to preferences on various measures of musical fidelity in the sound system. The issue is not difference in importance, but difference in character: different types of attributes require different trade-offs. A uniform monetary commodity is always traded as a commodity. Moreover, every objective is assumed to have a price.

3. A consequence of the equivalence between utility and a monetary commodity is that utility is relative:

... utility is a number up to a linear transformation.

We do not undertake to fix an absolute zero and an absolute unit of utility. [61] (p25)

Preference is absolute: a preference of zero is defined as unacceptable and a preference of one is defined as ideal. The absolute definition of zero preference is essential to design. The annihilation axiom for rational design decision-making relies on the absolute definition of zero preference. Failure to meet a minimum requirement in one aspect of the design must render the entire design unacceptable. Because there is no absolute zero of utility, there is no notion of absolute unacceptability in utility theory: a sufficiently high utility in another attribute can always compensate. This does not realistically represent design decisions.

Utilities are commonly aggregated with the arithmetic mean, which is a function in the family of weighted means:

$$(2.15) \quad \mathcal{P}_{s=1}((\mu_1, \omega_1), \dots, (\mu_N, \omega_N)) = \frac{1}{N} \sum_{i=1}^N \omega_i \mu_i$$

As discussed in Section 2.5, functions  $\mathcal{P}_{s>0}$  do not satisfy annihilation and hence are not design-appropriate. In general, aggregation functions in utility theory are not required to satisfy annihilation because utility is relative and zero utility does not represent any absolute notion of null preference or unacceptability.

### Matrix Methods

Design imprecision during the earliest stages of the design process is manifested as a multiplicity of alternative concepts. Morphological matrices [48] classify concepts by function, solution variant, working principle, type of motion, *etc.*, and in doing so, facilitate the generation of new concepts. Concepts are typically not sufficiently refined for quantitative analysis. Concept selection matrices [3, 48, 52] rank alternatives against evaluation criteria. Rankings are typically informally estimated against an existing design or some other datum. The weighted sum of rankings identifies promising alternatives. Pugh [52] also describes an alternative preliminary ranking scheme that has only three ratings: “+” (better than datum), “−” (worse than datum), and “S” (same as datum). These ratings are not summed algebraically, as in other methods, but rather serve to indicate the strengths and weaknesses of each alternative. It is in this respect that concept selection charts are most effective. They are not intended to be formal analyses. The summation of numeric rankings does not accurately reflect how criteria should be aggregated.

The analytic hierarchy process, or AHP [53], is a systematic procedure for determining the relationships between elements in a hierarchy of progressively more specific aspects of a problem. The axioms of the AHP do not include strong assumptions of rationality in decision-making [53]. The AHP combines attributes using a weighted sum which does not satisfy annihilation.

## Optimization

Optimization does not expressly model uncertainty. The purpose of design optimization is to algorithmically search for the “best” design relative to a single overall criterion. Papalambros and Wilde [49] identify four steps in the design optimization approach:

1. The selection of a set of variables to describe the design alternatives.
2. The selection of an objective (criterion), expressed in terms of the design variables, which we seek to minimize or maximize.
3. The determination of a set of constraints, expressed in terms of design variables, which must be satisfied by any acceptable design.
4. The determination of a set of values for the design variables, which minimize (or maximize) the objective, while satisfying all constraints.

In practice, steps 2 and 4 pose the greatest difficulty. It is not always possible to represent all relevant design requirements in a single objective. “The importance of optimization lies not in trying to find out all about a system, but in finding out, with the least possible effort, the best way to adjust the system. [2]” Ultimately, optimization does not seek to explore the design problem, but is interested only in obtaining the single “best” solution. This directed, point-based approach leads to algorithmic efficiency but is subject to two of French’s three criticisms:

- Are the algorithms and the criterion used to optimize the design an acceptable and rational emulation of the decision-maker?
- Is the process of design optimization itself informative, and does it guide the evolution of the decision-maker’s beliefs and preferences?

Engineers at one major U.S. automobile manufacturer tend to view optimization as a black box design tool [12]. Given constraints and a set of design variables, the optimization software simply searches for the design that is (locally) optimal with

respect to a given objective, *e.g.*, weight or stiffness. The software does not facilitate understanding of the design space other than at the single optimal design point. Moreover, the real problem has multiple objectives. Thus, optimization proceeds by iteratively cycling through several objectives. Occasionally engineers are tempted to “tweak” the final design in order to trade-off one objective for another, but this leads to designs that are non-optimal in some unexpected direction [12]. The problems of local optima, multiple objectives, and to a lesser extent lack of participation in the optimization process have been addressed in more advanced optimization methods, notably genetic algorithms [20] and various multi-objective optimization formulations [5, 19, 58]. Dlesk and Liebman [14] describe a multi-objective design methodology that also allows for uncertainty via “hedging,” a more systematic alternative to tweaking, and sensitivity analysis about the design point.

Optimization is fundamentally a point-based approach. Moreover, it emphasizes objectives as opposed to preferences on objectives, and thus implicitly assumes that preference is a simple, often monotonic function on each objective. Additionally, constraints are typically assumed to be precise. Yet these assumptions allow optimization algorithms to be computationally efficient and readily implemented.

### **Set-based Methods**

Ward *et al.* [64, 65], in a remarkable case study of Toyota’s design and development process, characterize a new approach that they refer to as “set-based concurrent engineering.”

Toyota designers think abouts sets of design alternatives, rather than pursuing one alternative iteratively. They gradually narrow the sets until they come to a final solution. [65] (p43)

To illustrate the power of set-based information relative to point-based information, Ward *et al.* [65] use a simple example problem: scheduling a meeting. A conventional point-by-point approach might begin with the meeting organizer picking a time and date. As other participants are contacted, the original time may turn out to be

unsatisfactory: a new time is picked but now the organizer must go back to check with all the people who were contacted previously. The new time may be unsatisfactory for them, requiring yet another change. For large, busy groups, this process quickly becomes time-consuming and unwieldy. There are two common strategies for shortening the search while retaining the point-by-point approach [65]. First, the group can meet briefly to decide when to have the meeting. This accelerates communication at the cost of the participants' time. For automobile development, this corresponds to collocating engineers that are working on the same project and requiring them to meet more often. Second, one or more powerful members of the group can dictate a time for the meeting, which is likely to produce a less than optimal solution, albeit quickly.

A third, set-based approach to planning a meeting requires all participants to submit the times that they are available, perhaps with preferences. A convenient time can quickly be found by taking the intersection of all sets of available times, a process now often automated. [65] (p44)

Ward *et al.* [65] observed five potential advantages to the set-based approach to design at Toyota:

1. "*Set-based concurrent engineering enables reliable, efficient communication.* [65]"

In a conventional, point-based approach, every design change can invalidate all previous decisions. Moreover, changes will not necessarily converge. Conversely, in a set-based approach, engineers communicate information that delineates the full set of possible designs. As the design process proceeds, this set is narrowed, supplementing without invalidating previous information. Set-based communication at Toyota appears to have several consequences. First, it eliminates work that is subsequently invalidated. "Toyota's body designers waste little time on detailed designs that cannot be manufactured because the manufacturing personnel can precisely define the set of bodies that are manufacturable . . . . [65]" Second, it reduces the number and length of meetings. "Toyota's engineers and suppliers can work relatively independently, because

each meeting communicates information about an entire set of designs. [65]” Toyota also achieves a high level of concurrency in its engineering process without collocating or dedicating its development teams. Third, set-based communication eliminates a major incentive to delay work. With a point-based approach, engineers downstream in the process may choose to delay making decisions because the design is subject to change. Toyota’s suppliers always know the amount of imprecision in their specifications and are therefore able to commit themselves accordingly. Finally, set-based communication can increase trust in working relationships. Informing a supplier about the set of possible changes instead of simply providing the minimum information encourages trust.

2. “*Set-based concurrent engineering allows for greater parallelism in the process, with more effective, early use of subteams.* [65]” In a set-based approach, downstream processes can become involved as soon as the set of possible designs is sufficiently refined. Manufacturing innovation that applies to a broad set of products may influence product design.
3. “*Set-based concurrent engineering bases the most critical, early decisions on data.* [65]” The earliest design decisions have the greatest impact on the ultimate quality and cost [9, 10], but these decisions are made with the least data [64], and moreover, data that is the least precise. Set-based methods allow Toyota engineers to explore the space of possible designs before making important decisions.
4. “*The set-based process promotes institutional learning.* Designers are notoriously resistant to documenting their work, partly because they sense that documentation is generally useless. [65]” Documenting a point-based design process provides directions from one specific starting point through one particular path, to the current, specific design. These directions are only useful to revisit the particular designs explored. At Toyota, team members systematically explore larger regions of the design space. Lessons-learned books record

the manufacturability of various body designs. In this way, designers have available to them a clear and up-to-date map of the space of manufacturable body designs, without even talking to a manufacturing engineer.

5. “*Set-based concurrent engineering allows for a search of globally optimal designs.* [65]” “Rapid inch-up” innovation can only find “local optima”: the best possible design based on the current fundamental concept. Set-based concurrent engineering can explore many different concepts and may potentially find better solutions based on radically different concepts.

These five advantages are a compelling motivation for all set-based methods, including the method of imprecision.

In formalizing their own set-based methodology, Ward and Seering [62, 63] have developed a theory to propagate intervals with associated labels such as **Limits** and **Operating-Region**. In relation to their work, the method of imprecision has focused not on the characterization of the nature of each interval (*i.e.*, its “label”), but rather on the representation of degrees of preference. In practice, the method of imprecision is manifestly set- and even interval-based (see Chapters 3 and 5). Indeed, the methods described in this thesis are well described by the third, set-based approach to planning a meeting quoted above. Fuzzy sets are, after all, a generalization of ordinary crisp sets.

Many researchers have used fuzzy sets to represent imprecision in decision-making outside of engineering design [6, 16, 24, 27, 40, 41, 72]. Most of these formulations are based on fuzzy “and” and “or” operators and are directed at modeling linguistic uncertainty and fuzzy logic. Although the design appropriate  $\mathcal{P}_{\min}$  and  $\mathcal{P}_{\Pi}$  aggregation functions are used to combine fuzzy sets, two classes of functions that do not in general satisfy the annihilation and idempotency axioms are commonly used for fuzzy decision-making: *t-norms* [36, 16] and *t-conorms*. T-norms are bounded above by min. T-conorms are bounded below by max.

Zimmerman and Sebastian [73, 74] and Müller and Thärigen [39] have applied fuzzy sets to engineering design. Given the basic equivalence of preferences as defined in the method of imprecision and membership as defined in fuzzy set theory,

their methods are similar to those described in this thesis. The fundamental choice of fuzzy sets to model design imprecision yields the same mathematical entities to quantify and manipulate imprecision. The method of imprecision is specifically directed at design decision-making in particular, and thus the more intuitive notion of preference replaces membership. The axioms that define design-appropriate aggregation function are also specific to design. These are a few of the distinctions between the two approaches that have arisen because of the relatively specific focus of the method of imprecision on engineering design. The work of Zimmerman and Sebastian [73, 74] has mainly been applied to configuration design.

A significant distinction of the method of imprecision is the inclusion of design preferences. The explicit representation of the customer’s indirect preferences anticipated by the designer is unique to the method. Design preferences, however, must be mapped from the DVS to the PVS, which is a non-trivial additional step.

## 2.9 Conclusions

The method of imprecision models the design problem in terms of two separate spaces: the *design variable space* (DVS) is the set of all design alternatives under active consideration, and the *performance variable space* (PVS) is the set of all quantified performances that are achievable by the designs in the DVS. Design variables  $d_1, \dots, d_n$  distinguish design alternatives that the designer considers to be distinct for the purpose of analysis. Other attributes of the design either are not under active consideration or cannot be directly specified. The set of design variables forms an  $n$  vector,  $\vec{d}$ , that distinguishes a particular design alternative in the DVS. Performance variables  $p_1, \dots, p_q$  quantify design performance for each design:  $p_j = f_j(\vec{d})$ . The mappings  $f_1, \dots, f_q$  can be any calculation or procedure to evaluate the performance of a design, but a performance attribute must be explicitly quantified to be a performance variable. The set of performance variables forms a  $q$  vector,  $\vec{p} = \vec{f}(\vec{d})$ , that specifies the quantified performances of a design  $\vec{d}$ .

Imprecision is represented through quantifying the customer’s direct and indirect

preferences on design and performance variables:

- Functional requirements  $\mu_{p_1}, \dots, \mu_{p_q}$  quantify the customer's direct preference on performance variables based on *performance considerations*: the quantified aspects of design performance represented by performance variables.
- Design preferences  $\mu_{d_1}, \dots, \mu_{d_n}$  quantify the customer's anticipated preference on design variables based on *design considerations*: the unquantified aspects of design performance not represented by performance variables.

The precise differentiation between design and performance variables, between design preferences and functional requirements, and between design and performance considerations, is a key contribution of this thesis.

The individual functional requirements and design preferences are aggregated into a single overall preference  $\mu_o$ . High overall preference identifies preferred and hence promising designs and performances. Five axioms for rational design decision-making were presented in Section 2.3: commutativity, monotonicity, continuity, annihilation, and idempotency. Of these five, annihilation is specific to design. Annihilation is necessary so that a design that is unacceptable in some aspect because it fails to meet a minimum requirement must also be judged to be unacceptable overall. Including importance weightings necessitates a redefinition of these axioms and the addition of a sixth axiom, self-normalization, which allows weights to be freely scaled by any strictly positive constant  $\lambda$ . This property is also useful where preferences are aggregated hierarchically. Aggregation functions that satisfy the axioms for rational design decision-making are termed *design-appropriate*. In Section 2.5, a family of hierarchically consistent, design appropriate aggregation functions were defined using the weighted root-mean-power family of functions. This family of functions  $\mathcal{P}_c$ , parameterized in  $c$ , allow the degree of compensation to be continuously varied from non-compensating ( $\mathcal{P}_{c=0} = \mathcal{P}_{\min}$ ) through partially compensating, fully compensating ( $\mathcal{P}_{c=1} = \mathcal{P}_{\Pi}$ ), and supercompensating, to maximally compensating ( $\mathcal{P}_{c=2} = \mathcal{P}_{\max'}$ ). This family of aggregation functions, which represent the model decision-maker in the method of imprecision, allow a broad range of degrees

of compensation and satisfy postulated axioms for rational design decision-making. They permit attributes to be weighted in importance and they support hierarchical aggregation. Few of the decision-making methods reviewed in Section 2.8 can match all of these claims (few methods have been specifically developed for design decision-making). Thus it is suggested that, provided the axioms of design-appropriateness adequately reflect the decision-maker's notion of rationality, the method of imprecision can indeed allow the decision-maker to define an aggregation hierarchy that acceptably models how the decision-maker might actually trade-off preferences.

The electric vehicle example in Section 2.7 demonstrated the modeling of a design problem, and in particular the process of identifying design variables, performance variables, and design considerations and constructing the design preference aggregation hierarchy. The elucidation of this process, in particular the construction of the design preference hierarchy, is the second key contribution of this thesis. The modeling of performance considerations as performance variables is not new and its suitability is not in debate. Whether preferences suitably model imprecision with respect to performance variables is, however, as yet unproven: this issue will be addressed in Chapter 5. But the two key issues here are first, whether the chosen means of representing and aggregating the customer's anticipated preferences on design variables is a suitable or even meaningful model of the design problem, and second, whether the suggested process for constructing the model is both feasible and informative.

The specific issues discussed as design considerations, such as aesthetics and manufacturability, are clearly relevant to design, yet they are difficult to include in any formal methodology. The work presented in this thesis not only allows the representation of these "soft" issues, but introduces a clearly defined formal structure for quantifying their consequences. Any relevant issue that can be related to a design variable can be modeled in this way. That design issues can be formed into a hierarchy with importance weightings is not controversial. The innovation of separating design considerations from performance considerations may at first make little sense, but since performance considerations will be explicitly evaluated and

their consequences calculated, only the design considerations remain to be quantified on the design variables. Using the designer's experience and judgement to project the customer's preferences back onto the design variables is already common practice. Indeed, a primary difficulty in implementing the method will be to selectively turn off the designer's automatic mapping of all of the customer's preferences onto design variables. Representing design considerations as the customer's anticipated preferences on design variables is therefore a suitable model: it is not inconsistent with how a designer thinks about design. Moreover, such a structure is also meaningful to the designer: it is readily interpreted.

That the process of identifying design considerations and constructing the design preference aggregation hierarchy is feasible has been demonstrated for one specific example. It is anticipated that this procedure will be feasible for any design problem for which the designer has a sufficient understanding. However, the author is clearly not an expert on electric vehicle design. Yet the process of enumerating design considerations, explicitly relating design considerations to design variables, constructing a hierarchy, determining relative importance and degree of compensation in aggregation, and examining the resulting model, forced a careful analysis of how the design variables impact design considerations, and as a result many important issues were clarified. This critical analysis of design considerations separately and in relation to design variables and an aggregation hierarchy is itself a valuable exercise. Thus it is suggested that the process of constructing the model is not only feasible and informative, but also requires the designer to more clearly distinguish and explicitly quantify the beliefs and preferences that are to be modeled.

## Chapter 3

### Calculating Imprecision

“Now,” said Rabbit, “this is a Search, and I’ve Organized it—”

“Done what to it?” said Pooh.

“Organized it. Which means — well, it’s what you do to a Search, when you don’t all look in the same place at once . . . .”

*A. A. Milne* (1882–1956), “The House at Pooh Corner”

Chapter 2 described how individual preferences can be hierarchically aggregated into an overall preference  $\mu_o$ . In implementing the method of imprecision, a key difficulty is that design preferences are specified on the DVS and functional requirements are specified on the PVS.  $f$  provides a forward mapping from the DVS to the PVS, but the backwards mapping from the PVS to the DVS is typically not available. Hence design preferences are first mapped onto the PVS. The mapped design preferences are then traded-off against functional requirements to obtain  $\mu_o(\vec{p})$ , the overall preference function on the PVS.  $\mu_o(\vec{p})$  represents the combined preferences of the designer and the customer, expressed in terms of design performance  $\vec{p}$ . In order to obtain  $\mu_o(\vec{d})$ , the overall preference on the DVS, functional requirements must be mapped back onto the DVS and traded-off against design preferences.  $\mu_o(\vec{d})$  identifies design configurations that are promising in terms of the combined preferences of the designer and the customer.

This chapter describes computational methods that have been developed in order

to perform the calculations described above. The practical difficulties of mapping preferences from the DVS to the PVS while achieving efficiency in function evaluations are discussed in Section 3.3. The methods developed utilize optimization (Section 3.2) and design of experiments (Section 3.4). Many of these methods have been implemented in a computer program, the Imprecise Design Tool, which will be described in Chapter 4.

### 3.1 The Level Interval Algorithm

After specifying design preferences  $\mu_{d_1}(d_1), \dots, \mu_{d_n}(d_n)$  and functional requirements  $\mu_{p_1}(p_1), \dots, \mu_{p_q}(p_q)$ , and identifying the appropriate hierarchy of trade-off strategies, the individual  $\mu_{d_i}(d_i)$  are aggregated to obtain  $\mu_d(\vec{d})$ , the combined design preference on the DVS.  $\mu_d(\vec{d})$  is then mapped onto the PVS, using the extension principle [71]:

$$(3.1) \quad \mu_d(\vec{p}) = \sup\{\mu_d(\vec{d}) \mid \vec{p} = \vec{f}(\vec{d})\}$$

where sup over the null set is defined to be zero.  $\mu_d(\vec{d})$  is the combined design preference on the DVS, as distinct from  $\mu_d(\vec{p})$ , the combined design preference on the PVS.  $\mu_d(\vec{p})$  is obtained by mapping  $\mu_d(\vec{d})$  onto the PVS.

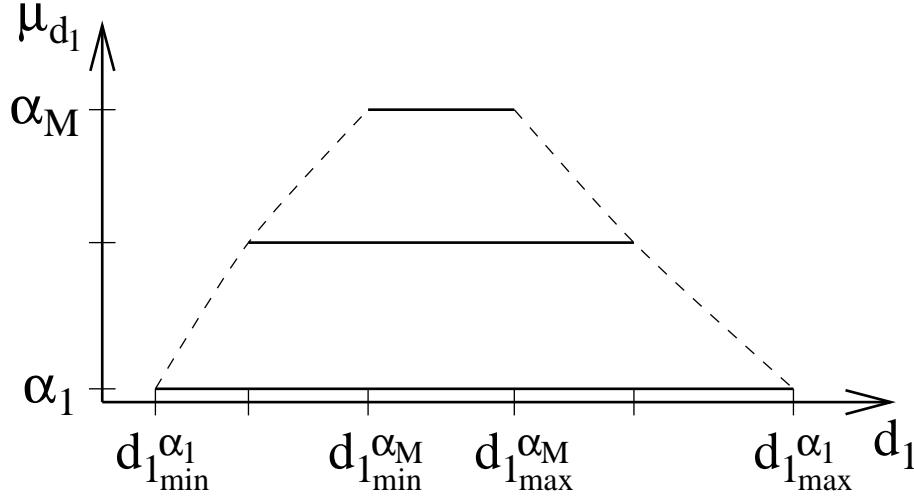
Individual functional requirements  $\mu_{p_1}, \dots, \mu_{p_q}$  are aggregated to obtain  $\mu_p(\vec{p})$ , the combined functional requirement on the PVS. The overall preference on the PVS is the aggregation of  $\mu_d(\vec{p})$  and  $\mu_p(\vec{p})$ :

$$(3.2) \quad \mu_o(\vec{p}) = \mathcal{P}(\mu_d(\vec{p}), \mu_p(\vec{p})).$$

$\mu_o(\vec{p})$  represents the combined preferences of the designer and the customer, expressed in terms of design performance.

In order to obtain  $\mu_o(\vec{d})$ , the overall preference on the DVS, the combined functional requirement  $\mu_p(\vec{p})$  is mapped onto the DVS:

$$(3.3) \quad \mu_p(\vec{d}) = \mu_p(\vec{p}) = \vec{f}(\vec{d}).$$



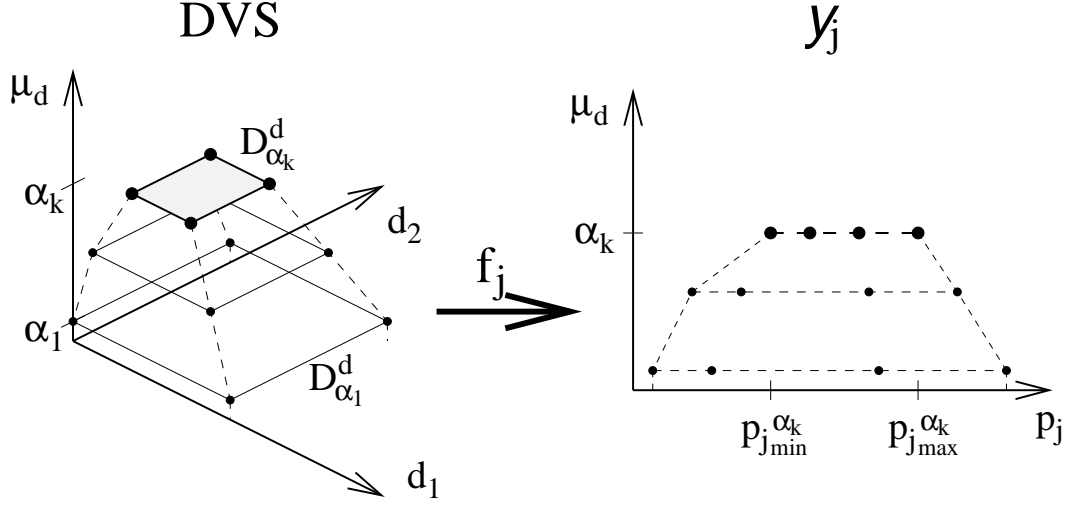
**Figure 3.1** Discretized design preference  $\mu_{d_1}$ .

Although  $\mu_p(\vec{d})$  is easily calculated using  $\vec{f}$  for any given design  $\vec{d}$ , determining how the preference function  $\mu_p(\vec{d})$  varies over sets of designs in the DVS is difficult without evaluating  $\vec{f}$  many times, especially because the inverse of  $\vec{f}$  is typically not available.

The first problem to be addressed is how to map design preference from the DVS to the PVS. Previously,  $\mu_d(\vec{p})$  has been calculated using the *Level Interval Algorithm*, or *LIA* [69], first proposed by Dong and Wong [15] as the “Fuzzy Weighted Average” algorithm and also called the “Vertex Method.” The LIA defines  $M$  levels of preference  $\alpha_1, \dots, \alpha_M$ . The individual design preference functions  $\mu_{d_i}(d_i)$  are discretized into  $\alpha$ -cut intervals  $[d_{i_{\min}}^{\alpha_k}, d_{i_{\max}}^{\alpha_k}]$  at these preference levels:

$$(3.4) \quad [d_{i_{\min}}^{\alpha_k}, d_{i_{\max}}^{\alpha_k}] = \{\mu_{d_i}(d_i) \geq \alpha_k\} \quad k = 1, \dots, M.$$

$\alpha$ -cut intervals for the design preference  $\mu_{d_1}$  are shown in Figure 3.1. These individual  $\alpha$ -cut intervals are then combined to obtain  $\alpha$ -cuts  $D_{\alpha_1}^d, \dots, D_{\alpha_M}^d$  in the DVS



**Figure 3.2** The Level Interval Algorithm.

which represent the combined design preference  $\mu_d(\vec{d})$ .

$$(3.5) \quad D_{\alpha_k}^d = \{\vec{d} \in \text{DVS} \mid \mu_d(\vec{d}) \geq \alpha_k\} \quad k = 1, \dots, M.$$

The LIA assumes that design preferences will be aggregated with a non-compensating trade-off and thus combines the individual  $\alpha$ -cuts by using the cartesian product:

$$(3.6) \quad D_{\alpha_k}^d = [d_{1_{\min}}^{\alpha_k}, d_{1_{\max}}^{\alpha_k}] \times \dots \times [d_{n_{\min}}^{\alpha_k}, d_{n_{\max}}^{\alpha_k}] \quad k = 1, \dots, M.$$

At the heart of the LIA is an enumerative procedure to map the combined design preference  $\alpha$ -cut  $D_{\alpha_k}^d$  onto individual intervals  $[p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}]$  in each  $\mathcal{Y}_j$ :

$$(3.7) \quad [p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}] = \{p_j(\vec{d}) \in \mathcal{Y}_j \mid \mu_d(\vec{d}) \geq \alpha_k\} \quad k = 1, \dots, M.$$

For each  $\alpha_k$ , the LIA evaluates  $p_j = f_j(\vec{d})$  for the  $2^n$  permutations of  $\alpha$ -cut end points which correspond to the corners of an  $n$ -cube defined by  $D_{\alpha_k}^d$  (there are  $n$  design variables and  $M$   $\alpha$ -cuts). Figure 3.2 illustrates how  $\alpha$ -cuts  $D_{\alpha_k}^d$  in two design variables  $d_1$  and  $d_2$  are mapped onto the interval  $[p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}]$ .  $f_j$  is evaluated

at the  $2^n = 4$  corner points of each  $D_{\alpha_k}^d$  rectangle. It is assumed that  $p_{j_{\min}}^{\alpha_k}$  and  $p_{j_{\max}}^{\alpha_k}$  will occur at these corner points, and not inside  $D_{\alpha_k}^d$ . Thus the minimum and maximum  $p_j$  among the four corner points defines the interval  $[p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}]$ . This is not true in general: the mapping  $f_j : \text{DVS} \rightarrow \mathcal{Y}_j$  and the combination function  $\mathcal{P}$  must satisfy certain conditions for the LIA to be exact [46]. In practice, these conditions require that  $f_j$  be monotonic: a severe restriction.

### 3.2 Optimization

The key limitation of the LIA, that it requires monotonicity, stems from the assumption that the extreme values of  $f_j$  will occur at the corner points of the  $D_{\alpha_k}^d$   $n$ -cube. The algorithm may thus be improved by relaxing this assumption [34]. The problem, restated, is to find:

$$(3.8) \quad \begin{aligned} p_{j_{\min}}^{\alpha_k} &= \min\{p_j = f_j(\vec{d}) \mid \vec{d} \in D_{\alpha_k}^d\} \\ p_{j_{\max}}^{\alpha_k} &= \max\{p_j = f_j(\vec{d}) \mid \vec{d} \in D_{\alpha_k}^d\}. \end{aligned}$$

Finding extrema within a subset of the DVS is a constrained optimization problem.

In choosing an optimization technique, a trade-off must be made between computational cost and robustness (*i.e.*, the ability to find the correct global extremum for various starting conditions). Traditional calculus-based optimization methods converge in relatively few function evaluations but seek only local minima. Randomized search methods such as genetic algorithms offer greater robustness [20] but require more function evaluations. Where function evaluations are relatively expensive, as is common in engineering design, traditional optimization methods are a pragmatic solution.

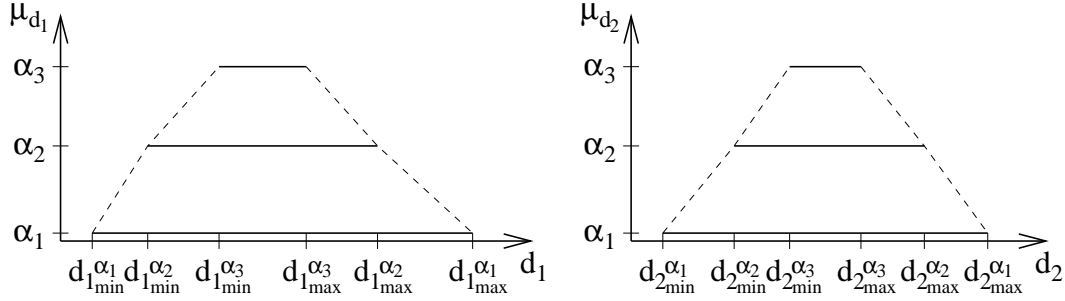
The computational implementation described in this thesis uses Powell's method, a calculus-based optimization algorithm that begins as a one at a time search. After each iteration a heuristic determines whether to replace the direction of maximum decrease with the net direction moved during the last iteration. This allows minimization down valleys while avoiding linear dependence in the set of search

directions [2]. Although optimization algorithms usually assume that variables are continuous, optimization can also be applied to discrete and mixed-discrete problems. In the aircraft engine design problem presented in Section 4.1, for example, all eight design variables are discrete.

An important feature for a practical computational tool is a means to trade-off the number of function evaluations against accuracy and reliability. Such an adjustment enables the designer to use the same program to obtain quick estimates as well as precise evaluations. This is implemented as a user-specified fractional precision that defines termination criteria for the optimization algorithm.

Suppose that it is necessary to incur the minimum number of function evaluations. A fractional precision of 1 would be specified, creating automatically satisfied termination criteria, and the optimization would proceed through exactly one iteration of a one at a time search using the maximum step size. The algorithm begins at one corner of the search space  $D_{\alpha_k}^d$ , and checks corners in each of the  $n$  directions given by  $d_1, \dots, d_n$ , moving to the minimum each time. It expends  $n + 1$  function evaluations to find each end point, and therefore  $2n + 2$  per  $\alpha$ -cut, as compared to  $2^n$  per  $\alpha$ -cut for the LIA. This is a substantial improvement, but the  $\alpha$ -cut interval obtained is only correct if  $f_j$  is monotonic: none of the interior points of the  $D_{\alpha_k}^d$   $n$ -cube are evaluated. Minimizing function evaluations in this way carries the cost of implicitly assuming monotonicity.

If  $f_j$  is known to be monotonic, this information can be used to further reduce the number of function evaluations. The first pass of the optimization algorithm identifies whether  $f_j$  increases or decreases in each  $d_i$ . Subsequent extrema can then be directly evaluated, without the need for searching. Hence where  $f_j$  is monotonic,  $n + 2$  function evaluations are required for the first  $\alpha$ -cut and 2 for each subsequent  $\alpha$ -cut.



**Figure 3.3** Design preference intervals for  $d_1$  and  $d_2$ .

### 3.3 Mapping Design Imprecision

In implementing the method of imprecision, a key step is mapping design preference  $\mu_d$  from the  $n$ -dimensional DVS to the  $q$ -dimensional PVS. If the individual design preferences  $\mu_{d_1}, \dots, \mu_{d_n}$  are to be combined with a non-compensating aggregation function  $\mathcal{P}_{\min}$ , the combined design preference  $\alpha$ -cuts  $D_{\alpha_1}^d, \dots, D_{\alpha_M}^d$  are given by the cartesian product of the individual design preference  $\alpha$ -cuts  $[d_{i_{\min}}^{\alpha_k}, d_{i_{\max}}^{\alpha_k}]$ , as in the LIA. The resultant  $D_{\alpha_1}^d, \dots, D_{\alpha_M}^d$  sets, which are  $n$ -cubes in the DVS, precisely describe the aggregation of individual preference intervals. But for aggregation functions other than  $\mathcal{P}_{\min}$ , the  $D_{\alpha_1}^d, \dots, D_{\alpha_M}^d$   $n$ -cubes do not fully describe the combined design preference  $\mu_d(\vec{d})$ . A two-dimensional example will illustrate the correct geometry of  $\mu_d(\vec{d})$ .

Figure 3.3 shows design preference intervals  $[d_{i_{\min}}^{\alpha_k}, d_{i_{\max}}^{\alpha_k}]$  at  $\mu_d = \alpha_1, \alpha_2, \alpha_3$  for two design variables  $d_1$  and  $d_2$ . Recall that each interval  $[d_{i_{\min}}^{\alpha_k}, d_{i_{\max}}^{\alpha_k}]$  defines the range of values for  $d_i$  over which the design preference  $\mu_{d_i}$  is at least  $\alpha_k$ . The combined design preference  $\mu_d$  obtained by aggregating these two discretized design preferences using an arbitrary aggregation function  $\mathcal{P}$  is shown (from above) in Figure 3.4. Three-dimensional views of  $\mu_d$  for  $\mathcal{P}_{\min}$  and  $\mathcal{P}_{\Pi}$  are depicted in Figures 3.5 and 3.6. Consider the center row, for which  $d_{2_{\min}}^{\alpha_3} \leq d_2 \leq d_{2_{\max}}^{\alpha_3}$  and therefore  $\mu_{d_2} = \alpha_3$  ( $\alpha_3$  is the highest preference). Where  $d_{1_{\min}}^{\alpha_1} \leq d_1 \leq d_{1_{\max}}^{\alpha_2}$ ,  $\mu_{d_1} \geq \alpha_1$  and thus  $\mu_d \geq \mathcal{P}(\alpha_1, \alpha_3) = \alpha_{1,3}$ . Similarly, where  $d_{1_{\min}}^{\alpha_2} \leq d_1 \leq d_{1_{\max}}^{\alpha_3}$ ,  $\mu_{d_1} \geq \alpha_2$  and

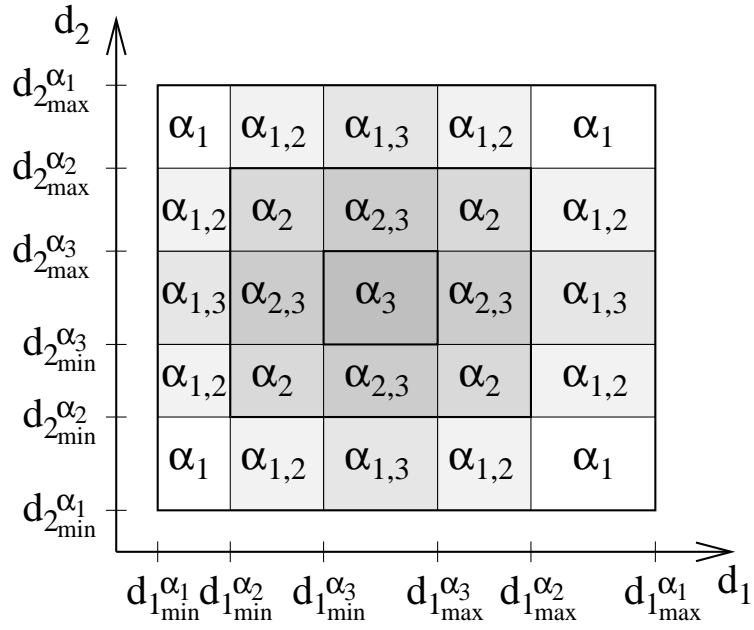


Figure 3.4 Combined design preference  $\mu_d = \mathcal{P}(\mu_{d_1}, \mu_{d_2})$ .  $\alpha_{i,j} = \mathcal{P}(\alpha_i, \alpha_j)$ .

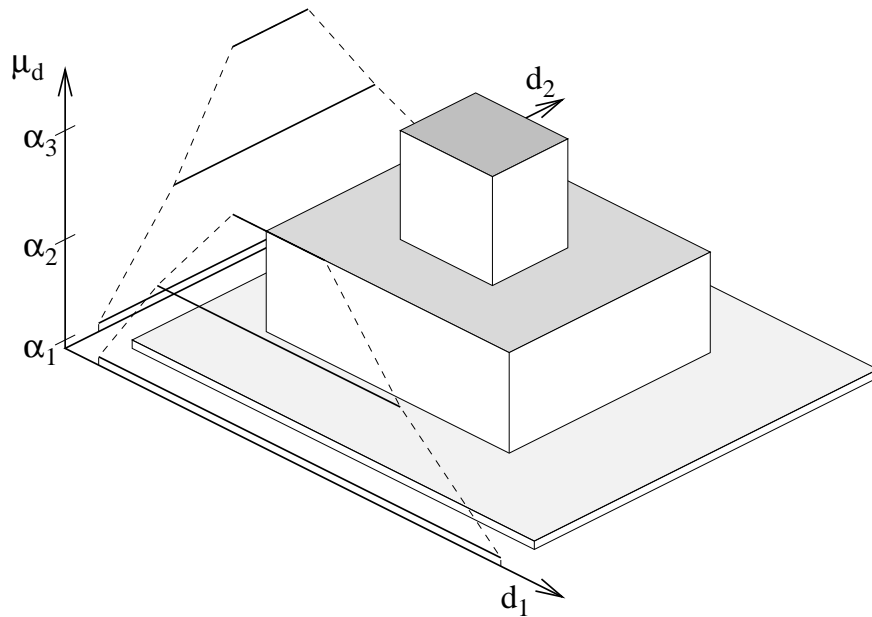
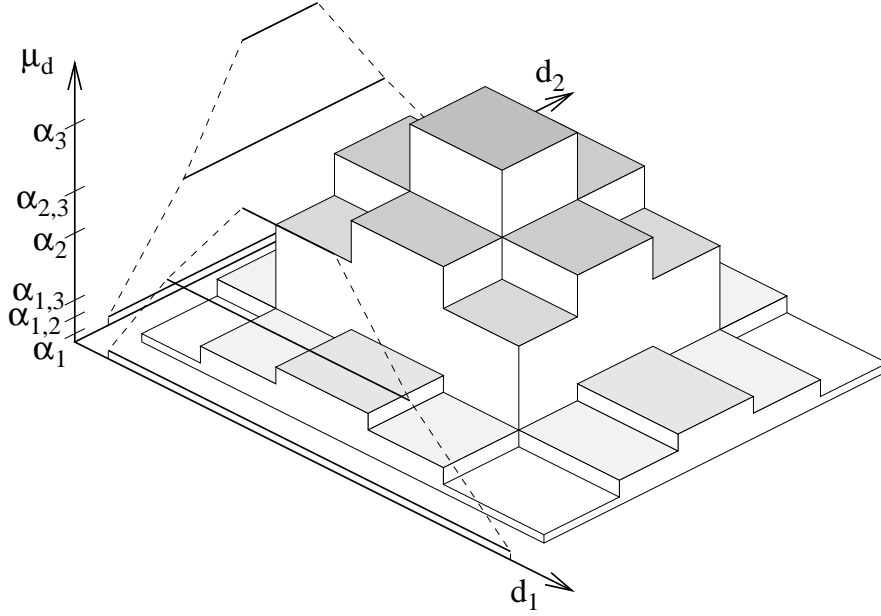


Figure 3.5 Combined design preference  $\mu_d = \mathcal{P}_{\min}(\mu_{d_1}, \mu_{d_2})$ .



**Figure 3.6** Combined design preference  $\mu_d = \mathcal{P}_\Pi(\mu_{d_1}, \mu_{d_2})$ .

$\mu_d \geq \mathcal{P}(\alpha_2, \alpha_3) = \alpha_{2,3}$ . Where  $d_{1\min}^{\alpha_3} \leq d_1 \leq d_{1\max}^{\alpha_3}$ ,  $\mu_{d_1} = \alpha_3$  but  $\mathcal{P}(\alpha_3, \alpha_3) = \alpha_3$  by idempotency, and thus  $\mu_d = \alpha_3$ . Note that monotonicity (Axiom 2.14) ensures that  $\alpha_{1,3} \leq \alpha_{2,3} \leq \alpha_3$ ,  $\alpha_{1,2} \leq \alpha_2 \leq \alpha_{2,3}$ , and  $\alpha_1 \leq \alpha_{1,2} \leq \alpha_{1,3}$ . Because  $\mathcal{P}_{\min}(\alpha_1, \alpha_2) = \mathcal{P}_{\min}(\alpha_1, \alpha_3) = \alpha_1$  and  $\mathcal{P}_{\min}(\alpha_2, \alpha_3) = \alpha_2$ , the discretized  $\mu_d(d_1, d_2)$  for  $\mathcal{P} = \mathcal{P}_{\min}$  is a Mayan (*i.e.*, rectangular, stepped) pyramid with three levels:  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  (Figure 3.5). Each level of the pyramid is a rectangular  $\alpha$ -cut  $D_{\alpha_k}^d = [d_{1\min}^{\alpha_k}, d_{1\max}^{\alpha_k}] \times [d_{2\min}^{\alpha_k}, d_{2\max}^{\alpha_k}]$ . Thus  $\mu_d(d_1, d_2)$  is precisely described by three such  $\alpha$ -cuts  $D_{\alpha_1}^d$ ,  $D_{\alpha_2}^d$ , and  $D_{\alpha_3}^d$ . For aggregation functions other than  $\mathcal{P}_{\min}$ ,  $\alpha_{2,3}$  rises above  $\alpha_2$ . The sides of the pyramid bulge outwards, although because of the discretization of preference, this is manifested as an increase in preference levels along each side instead of an outwards expansion (Figure 3.6). The rectangular  $\alpha$ -cuts  $D_{\alpha_1}^d$ ,  $D_{\alpha_2}^d$ , and  $D_{\alpha_3}^d$  remain valid as long as the steps of the pyramid do not overlap. In this example, the only possible overlap is if  $\alpha_{1,3} \geq \alpha_2$ . But to fully describe  $\mu_d(d_1, d_2)$  for aggregation functions other than  $\mathcal{P}_{\min}$ , additional non-rectangular level sets must be defined for  $\alpha_{1,2}$ ,  $\alpha_{1,3}$ , and  $\alpha_{2,3}$ . Thus the basic LIA must be further modified to accommodate

aggregation functions other than  $\mathcal{P}_{\min}$ . Additional, non-rectangular level sets must be defined to correctly represent the aggregation of the discretized individual design preferences  $\mu_{d_1}, \dots, \mu_{d_n}$  into the combined design preference  $\mu_d(\vec{d})$ .

Consider the problem of mapping just the rectangular  $\alpha$ -cuts  $D_{\alpha_k}^d$  from the DVS to the PVS. The methods developed below are easily extended to deal with non-rectangular, intermediate level sets. Each  $D_{\alpha_k}^d$  maps onto an  $\alpha$ -cut  $P_{\alpha_k}^d$  in the PVS via  $\vec{f}: \text{DVS} \rightarrow \text{PVS}$  ( $\vec{p} = \vec{f}(\vec{d})$ ).

$$(3.9) \quad P_{\alpha_k}^d = \{\vec{p} \in \text{PVS} \mid \mu_d(\vec{p}) \geq \alpha_k\}.$$

$P_{\alpha_k}^d$  will in general be neither a  $q$ -cube, nor even defined by straight edges. But given the need to minimize the number of function evaluations and the preliminary nature of the design information, the exact geometry of  $P_{\alpha_k}^d$  need not be calculated: an approximation is sufficient. Indeed, as  $q$ , the number of performance variables, increases beyond 2, there is little reason to pursue more accurate results that may be difficult or impossible to interpret, especially where  $\mu_d(\vec{d})$  is described by multiple  $n$ -cubic and non- $n$ -cubic level sets, as discussed above.

A straightforward extension of the LIA to deal with multiple performance variables would use a  $q$ -cube approximation to  $P_{\alpha_k}^d$  defined by the cartesian product of the individual intervals  $[p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}]$ :

$$(3.10) \quad P_{\alpha_k}^{d\Box} = [p_{1_{\min}}^{\alpha_k}, p_{1_{\max}}^{\alpha_k}] \times \dots \times [p_{q_{\min}}^{\alpha_k}, p_{q_{\max}}^{\alpha_k}].$$

This approximation is accurate only for severely restricted  $\vec{f}$ . Indeed,  $\vec{f}$  can only scale the  $n$ -cube  $D_{\alpha_k}^d$  in the principal  $p_j$  directions. There is also an implicit assumption that the  $p_j$ 's are independent, so that extrema can be independently determined. For these reasons  $P_{\alpha_k}^{d\Box}$  is an inadequate approximation.

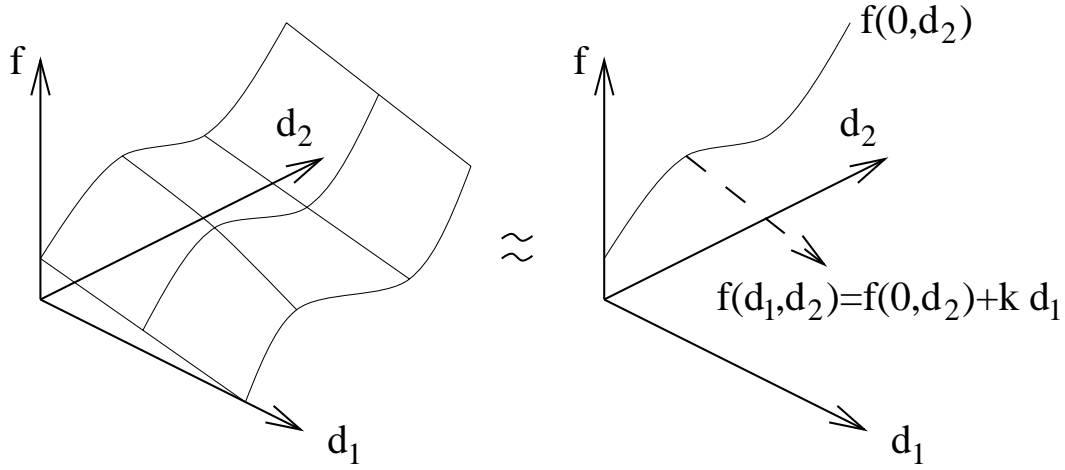
A superior approach is to selectively approximate  $\vec{f}$  as some simple function  $\vec{f}'$  over  $D_\epsilon^d$  (the  $\alpha$ -cut at infinitesimal  $\alpha = \epsilon$ , where  $0 < \epsilon \ll 1$ ). From Equation 3.5,  $\alpha$ -cuts with higher preference  $\alpha$  are subsets of  $\alpha$ -cuts with lower  $\alpha$ . Thus  $D_\epsilon^d$  contains all  $\alpha$ -cuts with non-zero  $\alpha$  and includes all potentially acceptable (non-zero

preference) design alternatives.  $D_{\alpha_k}^d$  can then be directly mapped onto the PVS, using  $\vec{f}'$ . A linear approximation is the obvious first choice:

$$\begin{aligned}
 \vec{f}'(\vec{d}) &= \begin{bmatrix} f'_1(\vec{d}) \\ \vdots \\ f'_q(\vec{d}) \end{bmatrix} = \vec{f}'(\vec{d}_{\text{ctr}}) + \vec{\Delta} + \mathbf{A}[\vec{d} - \vec{d}_{\text{ctr}}] \\
 (3.11) \quad &= \begin{bmatrix} f_1(\vec{d}_{\text{ctr}}) \\ \vdots \\ f_q(\vec{d}_{\text{ctr}}) \end{bmatrix} + \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_q \end{bmatrix} + \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{q1} & \dots & a_{qn} \end{bmatrix} \begin{bmatrix} d_1 - d_1^{\text{ctr}} \\ \vdots \\ d_n - d_n^{\text{ctr}} \end{bmatrix}
 \end{aligned}$$

where  $\vec{d}_{\text{ctr}}$  is the center point of  $D_{\alpha_k}^d$  and  $\Delta_j$  is the distance that  $f'_j$  is offset from  $f_j$  at  $\vec{d}_{\text{ctr}}$ . The elements  $a_{ji}$  of the matrix  $\mathbf{A}$  are linear regression coefficients that suitably approximate  $\vec{f}'$  over the entire search space  $D_{\alpha_k}^d$ . Although a linear approximation is not the only choice, higher order approximations introduce additional complexity, both in the shape of the level sets mapped onto the PVS and in the computational algorithm, that is not clearly justified. It is assumed that the cost of each function evaluation is not negligible and that gradient information is not readily available. Where these assumptions do not hold, other approaches may be applicable, such as continuation methods [26].

It is conjectured that if  $\vec{f}'$  is not strongly non-linear, a linear approximation will be adequate to sketch  $P_{\alpha_k}^d$  where the precise geometry is not required. During preliminary design, approximate answers are sufficient. As the design is refined, the set of design alternatives under consideration will be reduced in size. A linear approximation is likely to be more accurate over a smaller set of designs. The difficulty of interpreting an irregular  $P_{\alpha_k}^d$  set with curved boundaries in more than two dimensions suggests that a higher order approximation may be of limited value for problems with more than two performance variables. Yet it is important to consider where on  $P_{\alpha_k}^d$  more accurate results might be desirable. Although the detailed geometry of  $P_{\alpha_k}^d$  need not be known precisely, certain points on its boundary will be used to make design decisions, and these need to be determined with greater



**Figure 3.7** Reducing the search space with linear regression.

accuracy and reliability. The extremal points on the boundary of  $P_{\alpha_k}^d$  in each of the  $p_j$  directions fall into this category. These points are determined by optimization in the design variables that are not acceptably linear [32]. Acceptably linear variables are approximated by regression equations, thus shrinking the search space for optimization by one dimension (Figure 3.7).

A linear approximation also provides a simple measure of the sensitivity of each performance variable to changes in each design variable. This can be related to values of  $\alpha_k$  and normalized by the largest value separately for each performance variable to obtain a relative measure of design sensitivity at discrete levels of preference.

**Definition 3.1** The design sensitivity  $\kappa_{ji}^{\alpha_k}$  is the sensitivity of  $p_j$  to the variation in  $d_i$  defined by the  $\alpha$ -cut interval at  $\alpha = \alpha_k$ , relative to the largest value of  $\kappa_{ji}^{\alpha_k}$  for each  $p_j$  at the lowest  $\alpha_k$ :

$$(3.12) \quad \kappa_{ji}^{\alpha_k} = \frac{a_{ji}(d_{i\max}^{\alpha_k} - d_{i\min}^{\alpha_k})}{\max\{|\kappa_{j1}^{\alpha_1}|, \dots, |\kappa_{jn}^{\alpha_1}|\}}$$

where  $\alpha_1$  is the lowest value of  $\alpha_k$ . □

The design sensitivity  $\kappa_{ji}^{\alpha_k}$  is distinct from the  $\gamma$ -level measure [67], which measures the sensitivity of a performance variable to a design preference function exponentially weighted about a given preference  $\gamma$ .  $\kappa_{ji}^{\alpha_k}$  measures the sensitivity of a performance variable to a design preference function specifically at a preference  $\alpha_k$ , normalized with respect to the largest value of  $\kappa_{ji}^{\alpha_k} \forall k, i$  for each  $p_j$ .

For a particular performance variable  $p_j$ ,  $\kappa_{ji}^{\alpha_k}$  identifies which design variables have the greatest influence, and indicates the sign of the linear coefficient. By comparing  $\kappa_{ji}^{\alpha_k}$  for different performance variables, appropriate design variables can be chosen in order to, for example, reduce  $p_1$  and increase  $p_2$  simultaneously. Design sensitivities summarize the information contained in the linear regression coefficients.

A final benefit of constructing a linear approximation is that it provides a computationally tractable means to map the combined functional requirement from the PVS onto the DVS. The reverse mapping for  $\vec{f}$  is typically not available: given a performance  $\vec{p}$ , there is no direct means of determining its pre-image  $\{\vec{d} \mid \vec{p} = \vec{f}(\vec{d})\}$ . The linear approximation  $\vec{f}^{\vec{}}$ , however, can be reversed, though typically  $n > q$  (there will be more design variables than performance variables) and thus the pre-image of a single performance  $\vec{p}$  will be a set of points in the DVS. The pre-image  $\{\vec{d} \mid \vec{p} = \vec{f}^{\vec{}}(\vec{d})\}$  can be found by methods such as Gaussian elimination. The combined functional requirement  $\alpha$ -cut in the PVS is defined analogously to the combined design preference  $\alpha$ -cut in the DVS:

$$(3.13) \quad P_{\alpha_k}^p = \{\vec{p} \in \text{PVS} \mid \mu_p(\vec{p}) \geq \alpha_k\} \quad k = 1, \dots, M.$$

The combined functional requirement mapped onto the DVS is  $D_{\alpha_k}^p$ :

$$(3.14) \quad D_{\alpha_k}^p = \{\vec{d} \in \text{DVS} \mid \mu_p(\vec{f}(\vec{d})) \geq \alpha_k\}.$$

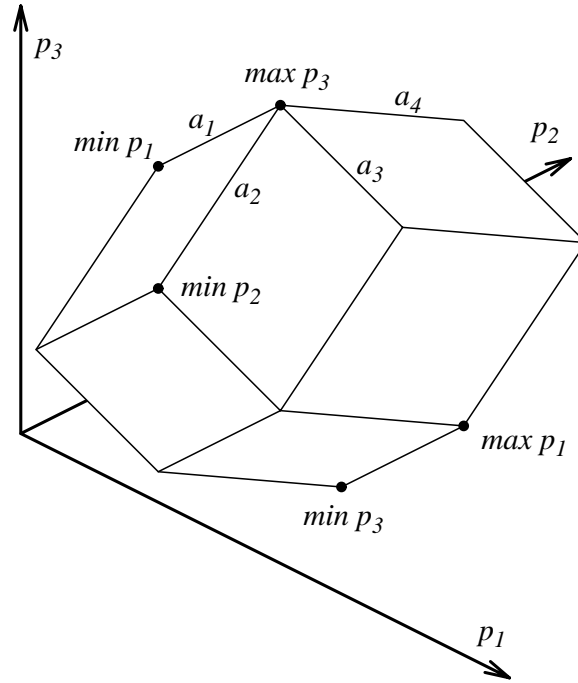
The pre-image of  $P_{\alpha_k}^p$  for  $\vec{f}^{\vec{}}$  approximates  $D_{\alpha_k}^p$ :

$$(3.15) \quad D_{\alpha_k}^{p'} = \{\vec{d} \in \text{DVS} \mid \mu_p(\vec{f}^{\vec{}}(\vec{d})) \geq \alpha_k\}.$$

Obtaining a linear approximation  $\vec{f}'$  fulfills four purposes: it removes acceptably linear design variables from the search space for optimization; it supplies a global approximation to  $\vec{f}$  over  $D_\epsilon^d$  for determining the geometry of  $P_{\alpha_k}^d$  between extremal points; it enables the calculation of design sensitivities  $\kappa_{ji}^{\alpha_k}$ ; and it provides a computationally tractable method to map preferences from the PVS onto the DVS. The mapping of  $D_{\alpha_k}^d$  onto the PVS does not, however, depend entirely upon the accuracy of the linear approximation  $\vec{f}'$ . The shape of  $P_{\alpha_k}^d$  in the PVS is estimated by obtaining extremal points in each  $p_j$  via optimization (facilitated by linear approximation), and then interpolating the bounding edges between points using  $\vec{f}'$ .

It is not expected that many performance variables will be well modeled by a linear approximation, even within a limited region of the DVS. But the linear approximation  $\vec{f}'$  furnishes additional information about the shape of  $P_{\alpha_k}^d$  away from extremal points that would otherwise be unavailable. Without  $\vec{f}'$ , the geometry of  $P_{\alpha_k}^d$  would only be known at extremal points.  $\vec{f}'$  is used to provide approximate information not to replace precise information, but to replace a lack of information. Useful information about  $P_{\alpha_k}^d$  can still be obtained even where  $f_1, \dots, f_q$  are all highly non-linear. Where  $\vec{f}$  is non-linear, the calculated shape of  $P_{\alpha_k}^d$  will be incorrect. But the extremal points obtained using optimization do not depend upon  $\vec{f}$  being linear. Moreover, even if  $\vec{f}'$  is completely unacceptable because  $f_1, \dots, f_q$  are all strongly non-linear in  $d_1, \dots, d_n$ ,  $P_{\alpha_k}^{d\Box}$  (Equation (3.10)) defined by the extremal values in each  $p_j$  provides a bounding set for  $P_{\alpha_k}^d$ .

Usually  $n \geq q$ : there will be at least as many design variables as performance variables. If  $n = q$  and  $\mathbf{A}$  is full rank,  $\vec{f}'$  maps the  $n$ -cube  $D_{\alpha_k}^d$  onto an  $n$ -parallelepiped in the PVS. Otherwise,  $\vec{f}'$  projects  $D_{\alpha_k}^d$  onto a  $q'$ -dimensional polyhedron where  $q' \leq q < n$ . This  $q'$ -dimensional polyhedron which is  $P_{\alpha_k}^{d'}$  is defined by the external surfaces of the projection of  $D_{\alpha_k}^d$  via  $\mathbf{A}$ . Since  $D_{\alpha_k}^d$  is an



**Figure 3.8** Approximated  $\alpha$ -cut  $P_\epsilon^{d'}$  on the PVS.

$n$ -cube, the directions of the parallel edges of  $P_{\alpha_k}^{d'}$  are given by the columns of  $\mathbf{A}$ :

$$(3.16) \quad \begin{bmatrix} a_{11} \\ \vdots \\ a_{q1} \end{bmatrix}, \dots, \begin{bmatrix} a_{1n} \\ \vdots \\ a_{qn} \end{bmatrix}.$$

Every bounding edge of  $P_{\alpha_k}^{d'}$  corresponds to an edge on  $D_{\alpha_k}^d$ , though some of the edges of  $D_{\alpha_k}^d$  map to the interior of  $P_{\alpha_k}^{d'}$ . As described above, optimization is used to more reliably calculate extremal points in each  $p_j$ . Modifying  $P_{\alpha_k}^{d'}$  to match these points will distort the geometry and edges may no longer be parallel. Additional accuracy may be obtained by explicitly calculating the remaining corner points on  $P_{\alpha_k}^{d'}$  which correspond to corners of the  $n$ -cube  $D_{\alpha_k}^d$  that are not extrema in any  $p_j$ .

### Example

Suppose that for a particular design problem, there are  $n = 4$  design variables

and  $q = 3$  performance variables. The designer wishes to conduct a preliminary analysis to explore how preferences on the design and performance variables intersect. Individual design preferences  $\mu_{d_1}, \mu_{d_2}, \mu_{d_3}, \mu_{d_4}$  are specified as intervals  $[d_{i\min}^{\alpha_k}, d_{i\max}^{\alpha_k}]$  at two preference levels:  $\alpha_1 = \epsilon, \alpha_2 = 1$ . An infinitesimal yet non-zero preference  $\epsilon$  indicates a barely acceptable variable value. The  $\alpha$ -cut interval at  $\alpha_1 = \epsilon$  identifies the largest acceptable interval of values for the variable. Values outside this range have zero preference and are thus unacceptable. Conversely, a preference of one indicates an ideal variable value. The  $\alpha$ -cut interval at  $\alpha_2 = 1$  identifies the ideal or target range of values for the variable. Specifying only two  $\alpha$ -cuts is a minimal implementation that is limited to calculating the boundaries of two sets: the set of acceptable designs and the set of ideal designs. Relatively few function evaluations are required. This is consistent with a preliminary analysis. For this minimal implementation, the distinction between aggregation functions is eliminated: at this level of discretization, the design preference intervals lead to the same  $n$ -cubic combined design preference  $\alpha$ -cuts  $D_\epsilon^d$  and  $D_1^d$  regardless of the aggregation functions used (this will be discussed in Section 5.1). These  $n$ -cubic  $\alpha$ -cuts fully describe the combined design preference  $\mu_d$  on the DVS.  $\vec{f}'$ , the linear approximation to the mapping  $\vec{f} : \text{DVS} \rightarrow \text{PVS}$  would be obtained by evaluating a central composite design over  $D_\epsilon^d$ . Suppose that the linear regression coefficients in the matrix  $\mathbf{A}$  have been obtained in this way:

$$(3.17) \quad \mathbf{A} = \begin{bmatrix} 1.1 & -0.12 & 2 & 0.68 \\ -0.1 & 1 & 0.24 & 2 \\ 1.1 & 0.88 & -1.24 & 1 \end{bmatrix}$$

$P_\epsilon^{d'}$ , the projection of  $D_\epsilon^d$  onto the PVS via  $\mathbf{A}$ , is shown in Figure 3.8.  $P_\epsilon^{d'}$  is a convex polyhedron that approximates the actual  $\alpha$ -cut  $P_\epsilon^d$ . Four edges that correspond to the four columns of  $\mathbf{A}$  are labeled  $a_1, \dots, a_4$ . These directions are the principal directions  $d_1, \dots, d_4$  mapped onto the PVS. The labeled corners are extrema in  $p_j$ : these points are obtained by optimization. The conventional optimization approach to this problem would be to search for the optimal  $p_1, p_2$ , or  $p_3$ , within a constrained

search space such as  $D_\epsilon^d$ . Thus if  $p_1$  is to be maximized and  $p_2$  and  $p_3$  are to be minimized, the information provided by optimization would be limited to three of the points in Figure 3.8 labeled max  $p_1$ , min  $p_2$ , and min  $p_3$ . Instead, the method of imprecision uses optimization to find both extrema in each  $p_j$  and then constructs an approximation to the entire set  $P_\epsilon^d$  using the linear approximation  $\vec{f}^j$ . The accuracy of the extrema is dependent only on the accuracy of the optimization algorithm used. Linear approximation is used to provide additional information: to fill in the gaps between extrema and paint a more complete picture.

$P_\epsilon^{d'}$  indicates the approximate region in the PVS within which  $\mu_d > 0$ . The performances  $\vec{p} \in P_\epsilon^{d'}$  correspond to the performances achievable by all designs that are at least minimally acceptable with respect to design considerations (the unquantified aspects of design performance not represented by performance variables). Functional requirements, which arise from performance considerations (the quantified aspects of design performance represented by performance variables) have not yet been included. Applying the combined functional requirement  $\mu_p(\vec{p})$  eliminates performances  $\vec{p} \in P_\epsilon^{d'}$  that are unacceptable because  $\mu_p(\vec{p}) = 0$ . The remaining subset of performances are acceptable relative to all specified preferences.

$P_1^{d'}$ , which approximates the region in the PVS within which  $\mu_d = 1$ , will be a subset of  $P_\epsilon^{d'}$ . The overall preference on the PVS,  $\mu_o(\vec{p})$ , is obtained by aggregating  $\mu_d(\vec{p})$  represented by these two  $\alpha$ -cuts with the combined functional requirement  $\mu_p(\vec{p})$ . The essential information given by  $\mu_o(\vec{p})$ , in this particular implementation, consists of the sets of performances achievable by ideal and acceptable designs:

- designs with  $\mu_o(\vec{d}) = 1$  are ideal, and
- designs with  $\mu_o(\vec{d}) \geq \epsilon$  are acceptable,

with respect to the preferences specified on the design and performance variables. Additionally, the effect of functional requirements on design performances is represented by the variation of  $\mu_o(\vec{p})$  between the extremes  $\epsilon$  and 1. Due to discretization, however, the combined design preference  $\mu_d$  is only mapped at the two extremes, and hence intermediate values of  $\mu_o(\vec{p})$  do not reflect the true variation of  $\mu_d(\vec{p})$ .

The combined functional requirement  $\mu_p(\vec{p})$  is mapped back onto the DVS using the linear approximation  $\vec{f}'(\vec{d})$ . The resulting functional requirement on the DVS,  $\mu_p(\vec{d})$ , is aggregated with the combined design preference  $\mu_d(\vec{d})$  to obtain the overall preference  $\mu_o(\vec{d})$ .  $\mu_o(\vec{d})$  identifies the ideal ( $\mu_o(\vec{d}) = 1$ ) and acceptable ( $\mu_o(\vec{d}) \geq \epsilon$ ) sets of designs described above.

### 3.4 Design of Experiments

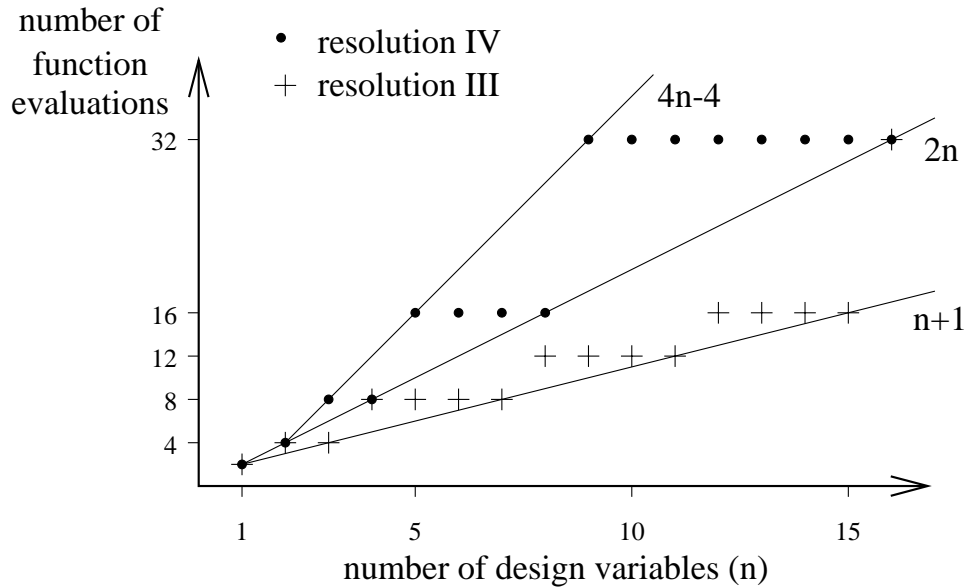
The linear approximations  $f'_1, \dots, f'_q$  are obtained using techniques adapted from statistical design of experiments. Design of experiments seeks to derive information about a process using as few observations as possible. It has two aims: to separate the effects to be measured from random noise, and to model the process with regression equations. The function  $f_j$  is treated as an unknown process. Note that if the process is deterministic, *e.g.*, a computer program, repeated evaluations will always give the same answer: the output contains no random noise. Therefore, statistical significance tests to distinguish the signal are unnecessary. This thesis discusses the use of experiment design only to model deterministic functions (though statistical significance tests are a valuable technique for processes subject to noise). The techniques used rely on orthogonal arrays, which specify an efficient, independent set of points at which the function is evaluated.

Orthogonal arrays are widely used not only for statistical design of experiments but also for the related Taguchi Method or Robust Design methodology [50, 51] and their direct application to engineering design is not new. Chi and Bloebaum describe a simple and practical application of orthogonal arrays to a material selection problem for multi-bar trusses in [8]. Korngold and Gabriele use experiment design to construct a global quadratic approximation for a multi-disciplinary problem [28]: their methods are similar to those that have been adopted here. A fundamental difference, however, is that Korngold and Gabriele have sought to solve a highly complex and general problem from a necessarily abstract and mathematical perspective. The work presented in this thesis seeks to apply experiment design techniques

specifically to facilitate method of imprecision calculations, and adopts a pragmatic approach that attempts to address the concerns of potential users. Engineers at one major U.S. automobile manufacturer, for example, consider each function evaluation to be a significant cost: it takes approximately 15 seconds on a supercomputer to evaluate a *simplified* finite element model of a vehicle structure. Furthermore, most engineers do not have the time to become experts on statistical techniques. New methodologies are expected to come pre-packaged as out-of-the-box software. Using experiment design to obtain linear regression models is efficient in function evaluations, does not require advanced statistical techniques, and is well suited to computer implementation.

The approach is essentially a *response surface method* [38], which seeks to optimize a response that is influenced by several variables. The function  $f_j$  is modeled over the search space  $D_\epsilon^d$  (the  $\alpha$ -cut at infinitesimal  $\alpha = \epsilon$ ). The Imprecise Design Tool uses a 2-level experiment design: two levels are sufficient to quantify linear effects. An additional center point checks for curvature: non-linearity of the function in the interior of the search space. A full factorial design would evaluate the same  $2^n$  corner points of  $D_\epsilon^d$  as the LIA, but since there are  $n$  main effects and 1 average to be determined, only  $n + 1$  evaluations are strictly necessary (excluding the center point). A fractional factorial design evaluates a balanced subset of corner points and is more efficient. But in reducing a full factorial experiment to a fractional factorial experiment, the  $2^n$  interactions between the  $n$  variables are unavoidably merged or *confounded* with each other, so that their effects cannot be distinguished. A key consideration is how interactions should be confounded. Main effects, which are to be measured, must not be confounded with other main effects. Moreover, it is desirable only to confound main effects with interactions that are unlikely to exist. It is assumed that main effects, due to a single variable (*e.g.*,  $d_1, d_2$ ), are more likely than two-way interactions (*e.g.*,  $d_1d_2, d_1^2$ ), which are in turn more likely than three-way (*e.g.*,  $d_1d_2d_3, d_1^2d_2, d_1^3$ ) and higher order interactions.

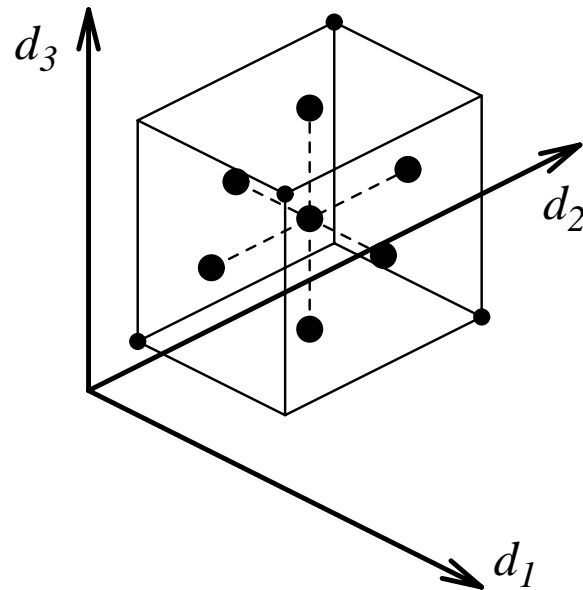
The *resolution* of an experiment design indicates the degree to which it confounds interactions. A resolution III design confounds main effects with two-way and higher



**Figure 3.9** Number of evaluations for a fractional factorial design.

order interactions, and thus satisfies the minimum requirement not to confound main effects. A resolution IV design confounds main effects with three-way and higher order interactions [4]. Resolution IV experiments provide more reliable information but require more observations.

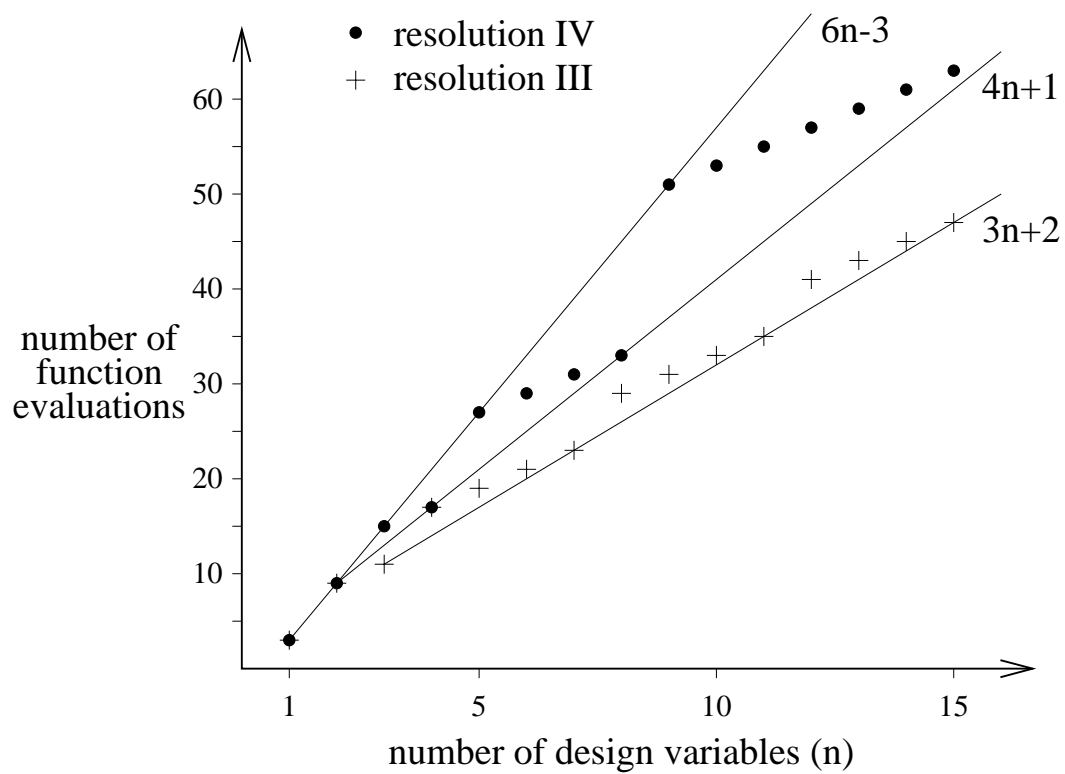
For  $n = 8$ , the smallest resolution IV design is a  $2^{8-4}$  fractional factorial design requiring  $2^4 = 16$  observations. The 4 in  $2^{8-4}$  indicates that the  $2^8$  full factorial design has been “folded” in half 4 times. A resolution III design would require 12 observations. Figure 3.9 compares the number of observations required for resolution III and IV designs using data from [51]. Resolution III designs approach the strictly necessary  $n + 1$  function evaluations. Resolution IV designs require between  $2n$  and  $4n - 4$  (where  $n > 1$ ) function evaluations. Of the 16 function evaluations required for the  $n = 8$  resolution IV design, 9 are strictly necessary to estimate the 8 main effects and 1 average, and so there are 7 “redundant” evaluations. But these evaluations are not necessarily wasted: they allow main effects to be separated from two-way interactions, and they provide 7 extra points to verify the accuracy of the linear regression model.



**Figure 3.10** Points evaluated in a central composite design.

The number of function evaluations can also be traded-off against accuracy for experiment design. The linear regression equations obtained replace the function where the approximation is acceptable (Figure 3.7). The criteria for “acceptable,” which determine how accurately the function is modeled, can be directly related to the user-specified fractional precision used by the optimization algorithm. This allows a single parameter to trade-off computational effort against accuracy for both optimization and experiment design.

A fractional factorial experiment only evaluates corner points. Thus comparison with the center point can only indicate whether  $f_j$  is non-monotonic and the degree to which it is non-linear, and cannot distinguish the design variable in which  $f_j$  is non-monotonic or non-linear. If  $f_j$  is non-linear in  $d_i$ ,  $f'_j$  will not accurately approximate  $\vec{f}$  in  $d_i$ : the approximation is still valid in  $d_i$  if this inaccuracy is within the user-specified precision. But if  $f_j$  is non-monotonic in  $d_i$ ,  $f'_j$  is not a valid approximation in  $d_i$ . Monotonicity in  $d_i$  is the minimum condition for  $d_i$  to be acceptably linear. In order to estimate non-linearity and non-monotonicity of  $f_j$  in each  $d_i$ , an additional “one-factor-at-a-time” experiment is conducted. Figure 3.10 shows the



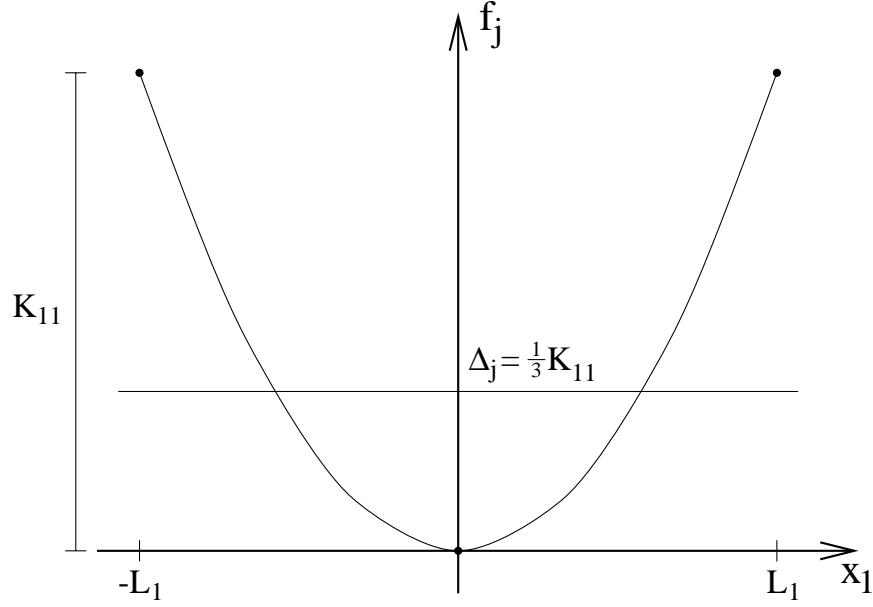
**Figure 3.11** Number of evaluations for a central composite design.

points that would be evaluated for a resolution III fractional factorial experiment (small dots) and a one-factor-at-a-time experiment (large dots) where there are three design variables. The combined experiment is termed a (*face-centered*) *central composite design* [38]. This arrangement was chosen in order to include corner points of the  $D_\epsilon^d$   $n$ -cube without evaluating points outside  $D_\epsilon^d$ . Extrema within  $D_\epsilon^d$  are likely to be corner points. In the absence of these constraints, other arrangements that seek to obtain a more balanced distribution of information over the search space are possible, *e.g.*, [38] and [21].

The number of function evaluations required for resolution III and IV central composite designs is indicated in Figure 3.11. If the function  $f_j$  is amenable to linear approximation, a maximum of  $4n + 1$  (resolution III) or  $6n - 3$  (resolution IV) evaluations will be incurred to obtain the regression equations and up to 2 evaluations will be required for the predicted  $\alpha$ -cut end points.  $4n + 1$  and  $6n - 3$  evaluations both exceed the  $n + 1$  evaluations required for a one at a time search, but the advantages are fourfold:

1. Monotonicity is not assumed: up to  $3n - 5$  “redundant” points test for monotonicity and linearity.
2. The center point tests for curvature.
3. The entire data set is used in estimating each effect, instead of two points.
4. An even distribution of corner points is sampled, instead of  $n + 1$  adjacent corners.

After calculating the linear regression matrix  $\mathbf{A}$ , the constant offsets  $\Delta_1, \dots, \Delta_q$  must be determined (see Equation (3.11)).  $f'_j(\vec{d})$  must approximate  $f_j(\vec{d})$  over the entire search space  $D_\epsilon^d$ . Setting  $\Delta_j = 0$  would give a Taylor approximation which is accurate near  $\vec{d}_{\text{ctr}}$  only. Setting  $\Delta_j$  such that  $f'_j(\vec{d})$  passes through the mean value of  $f_j(\vec{d})$  over all *evaluated* points would give a close approximation near the boundaries of  $D_\epsilon^d$  only, since only one interior point  $\vec{d}_{\text{ctr}}$  is evaluated. The Taylor expansion of  $f_j(\vec{d})$  near  $\vec{d}_{\text{ctr}}$  indicates that the residual error in approximating  $f_j$  is equal to the



**Figure 3.12** Minimizing  $\int_{-L_1}^{L_1} E^2 dx_1$  with offset  $\Delta_j$  ( $f_j = K_{11}x_1^2$ ).

offset  $\Delta_j$  plus second and higher order terms ( $\vec{x} = \vec{d} - \vec{d}_{\text{ctr}}$  where  $-L_i \leq x_i \leq L_i$ ,  $i = 1, \dots, n$ ):

$$\begin{aligned}
 E(\vec{x}) &= f'_j(\vec{d}_{\text{ctr}} + \vec{x}) - f_j(\vec{d}_{\text{ctr}} + \vec{x}) \\
 &= [a_{j1} \dots a_{jn}]\vec{x} + \Delta_j - f_j(\vec{d}_{\text{ctr}} + \vec{x}) \\
 (3.18) \quad &= \Delta_j - K_{11} \left(\frac{x_1}{L_1}\right)^2 - K_{22} \left(\frac{x_2}{L_2}\right)^2 - \dots - K_{12} \frac{x_1}{L_1} \frac{x_2}{L_2} - K_{13} \frac{x_1}{L_1} \frac{x_3}{L_3} - \dots
 \end{aligned}$$

For the purpose of determining an appropriate value for  $\Delta_j$ , assume that the error in approximating  $f_j$  is predominantly quadratic. In order to minimize  $\int_{D_\epsilon^d} E^2 dV$  (the square error integrated over  $D_\epsilon^d$ ),  $\Delta_j$  should be set to  $\frac{1}{3} \sum_{i=1}^n K_{ii}$  (Figure 3.12 illustrates a one-dimensional example). Qualitatively, it is clear that cross-terms ( $K_{ik}x_i x_k$  where  $i \neq k$ ) do not introduce the need to offset  $f'_j$  since they lead to an error that is anti-symmetric in  $x_i$  and  $x_k$ . Square terms ( $K_{ii}x_i^2$ ), however, do require a constant offset to minimize  $\int_{D_\epsilon^d} E^2 dV$  since they introduce an error in one direction only (positive or negative). The errors (for  $\Delta_j = 0$ ) at the  $N^\square$  corner

points  $\vec{d}_1^\square, \dots, \vec{d}_{N^\square}^\square$  evaluated by the fractional factorial experiment nominally average to  $-\sum_{i=1}^n K_{ii}$  (cross-terms cancel). The error at each point  $\vec{d}_i^-$  and  $\vec{d}_i^+$  evaluated in the  $d_i$  direction for the one-factor-at-a-time experiment is nominally equal to  $-K_{ii}$  (cross-terms equal zero). The sum of these errors is used to estimate  $\frac{1}{3} \sum_{i=1}^n K_{ii}$ :

$$\begin{aligned} \Delta_j &= \frac{1}{3(N^\square + 2)} \left( -\sum_{h=1}^{N^\square} E(\vec{d}_h^\square - \vec{d}_{\text{ctr}}) - \sum_{i=1}^n \left( E(\vec{d}_i^- - \vec{d}_{\text{ctr}}) + E(\vec{d}_i^+ - \vec{d}_{\text{ctr}}) \right) \right) \\ (3.19) &= \frac{1}{3(N^\square + 2)} \left( \sum_{h=1}^{N^\square} f_j(\vec{d}_h^\square) + \sum_{i=1}^n \left( f_j(\vec{d}_i^-) + f_j(\vec{d}_i^+) \right) - (N^\square + 2n) f_j(\vec{d}_{\text{ctr}}) \right) \end{aligned}$$

where  $E(\vec{x})$  is calculated for  $\Delta_j = 0$ . The points  $\vec{d}_1^\square, \dots, \vec{d}_{N^\square}^\square$  and  $\vec{d}_1^\pm, \dots, \vec{d}_n^\pm$  are symmetric on  $D_\epsilon^d$  such that the linear approximation  $f'_j$  averages to  $f_j(\vec{d}_{\text{ctr}})$  (for  $\Delta_j = 0$ ). Separating  $f_j(\vec{d}_{\text{ctr}})$  in Equation (3.19) simplifies the calculation.

As unacceptably non-linear design variables are eliminated from  $f'_j$ ,  $\Delta_j$  must be re-calculated since the square error in the design variable that was eliminated no longer needs to be offset.  $f_j$  will be calculated in all eliminated, non-linear directions and approximated only in the remaining, linear directions (Figure 3.7). The re-calculation of  $\Delta_j$  is complicated by the need to balance the data such that the result is not overly dependent on any single evaluated point. Every non-linear design variable that is eliminated reduces the space that  $f'_j$  approximates by one dimension. The points  $\vec{d}_i^-$  and  $\vec{d}_i^+$  evaluated in every remaining, linear direction ( $d_i$  such that  $i \in \mathcal{L}$ ) lie within this reduced space. But all of the fractional factorial corner points  $\vec{d}_1^\square, \dots, \vec{d}_{N^\square}^\square$  lie outside the reduced space. These corner points now provide less relevant information that must be adjusted: the  $K_{ii}$  such that  $i \notin \mathcal{L}$ , corresponding to eliminated design variables, must be subtracted from the estimate for  $\sum_{i=1}^n K_{ii}$  obtained from the error at each corner point. Moreover, each  $K_{ii}$   $i \notin \mathcal{L}$  can only be estimated from the errors at two evaluated points:  $\vec{d}_i^-$  and  $\vec{d}_i^+$ . Hence if  $f_j$  is not acceptably linear in all  $n$  design variables, a different formula must be

used to estimate  $\Delta_j$ :

$$(3.20) \quad \Delta_j = \frac{1}{3(2+n_{\mathcal{L}})} \left( \frac{2}{N^{\square}} \sum_{h=1}^{N^{\square}} f_j(\vec{d}_h^{\square}) - \sum_{i \notin \mathcal{L}} \left( f_j(\vec{d}_i^-) + f_j(\vec{d}_i^+) \right) + \sum_{i \in \mathcal{L}} \left( f_j(\vec{d}_i^-) + f_j(\vec{d}_i^+) \right) - 2n_{\mathcal{L}} f_j(\vec{d}_{\text{ctr}}) \right)$$

where  $f_j$  is acceptably linear in  $d_{i \in \mathcal{L}}$  and there are  $n_{\mathcal{L}}$  acceptably linear  $d_{i \in \mathcal{L}}$ .

A special case arises when there is only one remaining linear design variable for  $f_j$ , *i.e.*,  $n_{\mathcal{L}} = 1$ . Since  $f'_j$  will only be used to approximate  $f_j$  in a single design variable  $d_{i \in \mathcal{L}}$ ,  $\Delta_j$  can be directly estimated from  $\vec{d}_{i \in \mathcal{L}}^-$  and  $\vec{d}_{i \in \mathcal{L}}^+$  only, as in Figure 3.12:

$$(3.21) \quad \Delta_j = \frac{1}{6} \left( f_j(\vec{d}_{i \in \mathcal{L}}^-) + f_j(\vec{d}_{i \in \mathcal{L}}^+) - 2f_j(\vec{d}_{\text{ctr}}) \right).$$

### 3.5 Conclusions

In the method of imprecision, design preferences  $\mu_{d_i}$  are specified on design variables and functional requirements  $\mu_{p_j}$  are specified on performance variables. Individual design preferences  $\mu_{d_i}(d_i)$  are aggregated into the combined design preference  $\mu_d(\vec{d})$  and individual functional requirements  $\mu_{p_j}(p_j)$  are aggregated into the combined functional requirement  $\mu_p(\vec{p})$ . The aggregation of these combined preferences into the overall preference  $\mu_o$  is complicated by the need to map the combined design preference onto the PVS to obtain  $\mu_o(\vec{p}) = \mathcal{P}(\mu_d(\vec{p}), \mu_p(\vec{p}))$  or to map the combined functional requirement onto the DVS to obtain  $\mu_o(\vec{d}) = \mathcal{P}(\mu_d(\vec{d}), \mu_p(\vec{d}))$ . Typically, only the forward mapping  $\vec{f}: \text{DVS} \rightarrow \text{PVS}$  is available.

Previously, design preferences were mapped onto the PVS using the Level Interval Algorithm (LIA). The LIA begins by discretizing individual design preferences into  $M$  levels of preference  $\alpha_1, \dots, \alpha_M$ . The individual design preference intervals obtained are then aggregated into combined design preference  $\alpha$ -cut sets  $D_{\alpha_1}^d, \dots, D_{\alpha_M}^d$ . The LIA has four important limitations:

1. The  $\alpha$ -cuts  $D_{\alpha_k}^d$  must be  $n$ -cubes, which is accurate only if aggregation uses  $\mathcal{P}_{\min}$ .
2. The calculated performance variable endpoints  $p_{j_{\min}}^{\alpha_k}$  and  $p_{j_{\max}}^{\alpha_k}$  are only correct under certain conditions: in practice these require that  $f_j$  be monotonic.
3. The endpoints obtained only indicate extremal points in each  $p_j$  on  $P_{\alpha_k}^d$ , the combined design preference  $\alpha$ -cut in the PVS: the full geometry of  $P_{\alpha_k}^d$  is not determined.
4. Up to  $2^n$  function evaluations are required to evaluate each  $\alpha$ -cut, a number that quickly becomes prohibitive as the number of design variables  $n$  increases.

The methods presented in this chapter were developed to address these limitations.

The key limitation of the LIA, that it requires monotonicity, may be removed by reformulating the problem as a constrained optimization:  $p_{j_{\min}}^{\alpha_k}$  is the minimum and  $p_{j_{\max}}^{\alpha_k}$  is the maximum of  $p_j$  in  $D_{\alpha_k}^d$ . In the interests of minimizing the number of potentially expensive function evaluations, a traditional calculus-based optimization algorithm, Powell's method, was chosen for the computational implementation described in this thesis.

A key contribution of this thesis is the provision of a fractional precision that permits the designer to trade-off the number of function evaluations against the quality, *i.e.*, accuracy and reliability, of the answer obtained. This adjustment allows the designer to use the same computer program to obtain quick estimates as well as precise evaluations. Applying Powell's method with a maximum fractional precision of 1 reduces the number of function evaluations required per  $\alpha$ -cut from  $2^n$  to  $2n + 2$ , or  $n + 2$  if monotonicity is assumed. Such a minimalist computation is, however, unlikely to be robust.

For aggregation using functions other than  $\mathcal{P}_{\min}$  non- $n$ -cubic  $\alpha$ -cuts at intermediate levels of preference between  $\alpha_k$  may be created in addition to the  $n$ -cubic  $\alpha$ -cuts at preferences  $\alpha_k$  (Section 3.3). In Section 5.1 this complicated geometry will be shown to be simplified when the method is implemented.

In order to address the issue of robustness as well as the remaining two limitations of the LIA (that the  $D_{\alpha_k}^d$  must be  $n$ -cubes and that the full geometry of  $P_{\alpha_k}^d$  is not determined), an approximation  $\vec{f}^l$  for  $\vec{f} : \text{DVS} \rightarrow \text{PVS}$  is constructed over  $D_\epsilon^d$  (the  $\alpha$ -cut at infinitesimal  $\alpha = \epsilon$ ). Although Section 3.3 only discusses the problem of mapping  $n$ -cubic  $\alpha$ -cuts from the DVS to the PVS using  $\vec{f}^l$ , there is no fundamental difficulty in extending this approach to map non- $n$ -cubic  $\alpha$ -cuts. A linear approximation is chosen for  $\vec{f}^l$  for three reasons:

1. Higher order approximations introduce additional complexity, both in the shape of the level sets mapped onto the PVS and in the computational algorithm, that is not clearly justified.
2. If  $\vec{f}$  is not strongly non-linear, the selective use of a linear approximation to sketch  $P_{\alpha_k}^d$  where the precise geometry is not required is expected to be adequate: during preliminary design, approximate answers are sufficient.
3. The difficulty of interpreting an irregular  $P_{\alpha_k}^d$  set with curved boundaries in more than two dimensions suggests that a higher order approximation may be of limited value for problems with more than two performance variables.

A linear approximation is unlikely to accurately approximate  $\vec{f}$  in all  $n$  design variables. That is not, however, the aim of constructing  $\vec{f}^l$ . Obtaining a linear approximation  $\vec{f}^l$  fulfills four purposes:

1. It removes acceptably linear design variables from the search space for optimization.
2. It supplies a global approximation to  $\vec{f}$  over  $D_\epsilon^d$  for determining the geometry of  $P_{\alpha_k}^d$  between extremal points.
3. It enables the calculation of design sensitivities  $\kappa_{j_i}^{\alpha_k}$ .
4. It provides a computationally tractable, albeit approximate, means to map the combined functional requirement from the PVS onto the DVS.

The mapping of  $D_{\alpha_k}^d$  onto the PVS does not depend entirely upon the accuracy of the linear approximation  $\vec{f}'$ . The shape of  $P_{\alpha_k}^d$  in the PVS is constructed by obtaining extremal points in each  $p_j$  via optimization (facilitated by linear approximation), and then interpolating the bounding edges between points using  $\vec{f}'$ .

It is not expected that many performance variables will be well modeled by a linear approximation, even within a limited region of the DVS. But the linear approximation  $\vec{f}'$  furnishes additional information about the shape of  $P_{\alpha_k}^d$  away from extremal points that would otherwise be unavailable.  $\vec{f}'$  is used to provide approximate information not to replace precise information, but to replace a lack of information. It will be demonstrated in Section 4.2 that a selectively applied linear approximation is surprisingly effective in evaluating an example finite element model of an automobile body.

The linear approximation  $\vec{f}'$  over  $D_\epsilon^d$  is obtained using techniques adapted from experiment design. The use of experiment design to explore the design space and optimization assisted by linear approximation to map preferences is a key contribution of this thesis. Using experiment design to obtain a linear regression model is efficient in function evaluations, does not require advanced statistical techniques, and is well-suited to computer implementation. The Imprecise Design Tool uses a 2-level fractional factorial experiment design which evaluates a balanced subset of corner points: two levels are sufficient to quantify linear effects. An additional center point checks for non-linearity of the function in the interior of the search space. Resolution IV fractional factorial experiments require more function evaluations than resolution III experiments, but the additional evaluations increase the reliability of the information obtained. This will be demonstrated in Section 4.2.

Fitting a linear approximation  $f'_j$  to a function  $f_j$  over the entire search space  $D_\epsilon^d$ , instead of at a single point, necessitates an offset  $\Delta_j$ . A procedure for determining appropriate values for  $\Delta_1, \dots, \Delta_q$  in order to minimize  $\int_{D_\epsilon^d} E^2 dV$ , the square error integrated over  $D_\epsilon^d$ , was described in Section 3.4.

The fractional precision linked to optimization also controls the criteria for whether  $\vec{f}'$  approximates  $\vec{f}$  sufficiently accurately in each design variable  $d_i$ . Only

sufficiently linear variables are approximated for optimization. In order to discern the linearity of individual design variables, an additional one-factor-at-a-time experiment is conducted to complete a central composite design. The total number of function evaluations required is  $3n + 2$  to  $4n + 1$  for resolution III and  $4n + 1$  to  $6n - 3$  for resolution IV. These function evaluations are in addition to those required for subsequent optimization, but they are necessary for constructing the linear approximation  $\vec{f}'$ , and they provide robustness. Before optimization begins the search space is characterized and a balanced set of points is evaluated, although only the center point lies in the interior of  $D_\epsilon^d$ . The likely location of the global minimum based on evaluated points is chosen as the starting point for optimization. Thus experiment design is a valuable exploration of the search space that facilitates the subsequent optimization process.

## Chapter 4

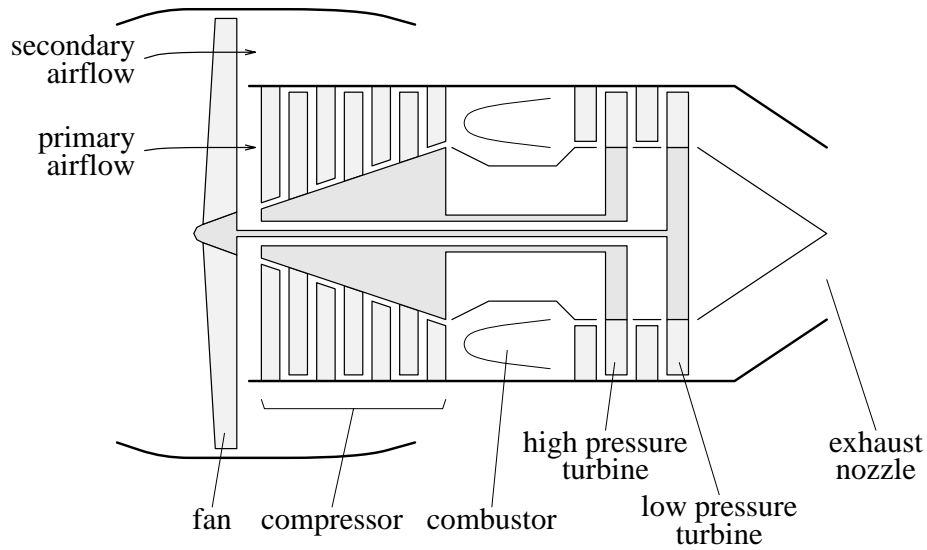
### Computational Implementation

The Imprecise Design Tool (IDT) is a C program developed to verify the algorithms described in Chapter 3 and to demonstrate the method of imprecision on engineering problems. The work described in this chapter addresses many of the practical difficulties that arose from trying to create a computational design tool that might be useful to engineers in industry. This section describes the evolution of the IDT itself, through example applications to aircraft engine development and automobile structure design.

#### 4.1 The Engine Development Cost Estimator

The first application of the IDT was to a cost estimation problem provided by General Electric Aircraft Engines, Cincinnati, Ohio. “Gas turbine engines exert a dominant influence on aircraft performance and must be designed for each specific application [35].” At General Electric, the Engine Development Cost Estimator (EDCE) is used to provide an early estimate of the cost of developing a new aircraft engine: a cost measured in hundreds of thousands of dollars. The EDCE is one of several programs that together estimate the total lifetime cost of an engine.

The EDCE uses a separate database for each engine program that contains estimates for the various components of development cost. Costs are modified by parameters such as learning, program length, and whether various tests are to be included in the development cost. The most important input variables which have

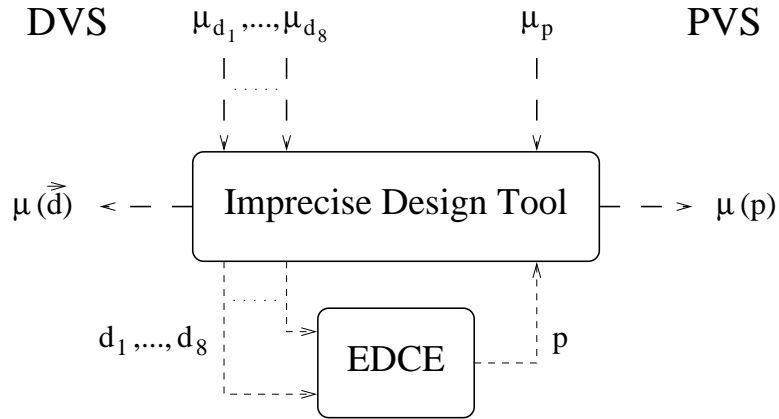


**Figure 4.1** Schematic diagram of a turbofan engine.

the greatest effect on the calculated cost represent the degree of innovation in the components and subsystems of the new engine. These variables are significantly imprecise: the degree of innovation for a particular component of the engine will, in general, not be known precisely in advance. Eight of the variables corresponding to eight subsystems were chosen to be design variables (Figure 4.1):

- $d_1$  control and accessories
- $d_2$  exhaust nozzle
- $d_3$  bearing and lubrication
- $d_4$  low pressure turbine
- $d_5$  high pressure turbine
- $d_6$  high pressure compressor
- $d_7$  combustor
- $d_8$  fan system

The EDCE represents degree of innovation as a percent change relative to an existing baseline design. A value of “0%” for  $d_8$ , which corresponds to the fan system, indicates that the engine to be developed does not possess a fan system. A value of “10%” indicates that only support engineering will be required. At the other



**Figure 4.2** The Imprecise Design Tool and the EDCE.

extreme, “200%” indicates a new fan with similar or existing technology, fitted to a new engine design. The numeric values of percent change for each of the ten levels defined by the EDCE are unimportant: the designer relies on the verbal definition of each level, which is specific to each input. Intermediate values between levels are undefined, and hence the eight inputs are effectively discrete. The EDCE produces a single output  $p = f(\vec{d})$ : the estimated development cost for the new engine.

The EDCE calculates development cost given a set of precisely specified, crisp inputs. The role of the IDT was to provide a fuzzy interface to the EDCE that quantified the inherent imprecision in the design variables  $d_1, \dots, d_8$  and performance variable  $p$  and evaluated imprecise outputs  $\mu(\vec{d})$  and  $\mu(p)$  (Figure 4.2). The EDCE defined the crisp mapping  $f : DVS \rightarrow PVS$  used by the IDT to perform fuzzy calculations.

Since the design variables  $d_1, \dots, d_8$  are discrete, each design preference  $\mu_{d_i}$  was specified as an array of preference values, one for each valid design variable value  $d_i \in \mathcal{X}_i$ . Since the IDT discretized preference into ten  $\alpha$  levels 0.1, 0.2, ..., 1.0 the user was effectively restricted to those ten preference values. The functional requirement on development cost was specified as a list of points  $(p, \mu_p)$ , defining a piecewise linear preference function.

This early version of the IDT assumed that all preferences would be traded-off with a non-compensating aggregation function  $\mathcal{P}_{\min}$ . Hence the individual design preferences  $\mu_{d_i}(d_i)$  could be aggregated into  $n$ -cubic combined design preference  $\alpha$ -cuts  $D_{\alpha_k}^d$  using the Cartesian product (Section 3.3). The  $D_{\alpha_k}^d$  were then mapped to intervals  $[p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}]$  on the PVS using the Level Interval Algorithm (LIA) described in Section 3.1. The LIA was valid because development cost was monotonic in each of the eight design variables, over the range of values specified. The representation of preference as intervals approximated the discrete DVS. Although the limits of the  $[d_{i_{\min}}^{\alpha_k}, d_{i_{\max}}^{\alpha_k}]$  and  $[p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}]$  intervals corresponded to valid points in the DVS, only certain discrete values had meaning within these intervals. Nevertheless, the calculation of seemingly continuous performance intervals was valuable in identifying the limits of development cost at each preference level  $\alpha_k$ . In the example presented below, the points identified by the IDT as promising were all interval endpoints that corresponded to valid points in the discrete DVS.

The original LIA required  $2^n = 256$  function evaluations to calculate each  $\alpha$ -cut interval in  $p$ . Ten  $\alpha$ -cuts required 2560 evaluations. At 15 seconds per evaluation on a Sun4, the total time required exceeded ten hours. A number of pragmatic measures were taken to substantially reduce the time required. A lookup table was created to store values of  $f(\vec{d})$  calculated by the EDCE. Using the lookup table avoided repeated function evaluations not only for subsequent iterations but also for the current design calculation. Because the design variables were discrete, adjacent  $\alpha$ -cut intervals often had common endpoints. These common endpoints only needed to be evaluated or looked up once. Furthermore, it became clear that specifying ten distinct  $\alpha$ -cuts was unwieldy and that restricting preferences to some subset of the values 0.1, 0.2, ..., 1.0 simplified preference calculations without sacrificing any important information.

The combined design preference on the PVS,  $\mu_d(p)$ , and the functional requirement  $\mu_p(p)$  were then aggregated using  $\mathcal{P}_{\min}$  to obtain the overall preference  $\mu_o(p)$ , expressed as an ordered list of pairs  $(p, \mu_o)$  defining a piecewise linear preference function on the PVS. The same step also identified the peak preference  $\mu_o^*$  and the

corresponding peak preference set of development costs  $\mathcal{Y}^*$ .

An early solution to the problem of mapping  $\mathcal{Y}^*$  back onto the DVS to obtain  $\mathcal{X}^*$  used the  $\alpha$ -cut calculations that mapped  $\mu_d(\vec{d})$  onto  $\mu_d(p)$ . The IDT first determined  $\alpha^*$ , the largest  $\alpha_k \leq \mu_o^*$ . For a non-compensating trade-off, the set of peak preference designs  $\mathcal{X}^*$  is a subset of  $D_{\alpha^*}^d$  because  $\mu_o = \min(\mu_d, \mu_p) \leq \mu_d \forall \mu_p$  and thus  $\mu_o(\vec{d}) < \alpha^* \leq \mu^* \forall \vec{d} \notin D_{\alpha^*}^d$ : only design configurations  $\vec{d} \in D_{\alpha^*}^d$  can have overall preference  $\mu_o(\vec{d})$  equal to  $\mu_o^*$ . Therefore,  $\mu(\vec{d})$  was calculated at every  $\vec{d} \in D_{\alpha^*}^d$  for which there existed a lookup table entry for  $f(\vec{d})$ . Any  $\vec{d} \in D_{\alpha^*}^d$  with  $\mu(\vec{d}) = \mu^*$  indicated a peak preference design configuration. Where  $f(\vec{d})$  was not immediately available,  $\mu_d(\vec{d})$  provided an upper bound since, for a non-compensating trade-off,  $\mu_o \leq \mu_d$ . Thus  $\mu(\vec{d})$  or an upper bound for  $\mu(\vec{d})$  was obtained at every  $\vec{d} \in D_{\alpha^*}^d$  that could potentially be a peak preference design configuration. Moreover, no additional function evaluations were required to obtain this information. In order to permit the user to visualize the variation of  $\mu_o$  on the DVS, points could be specified about which the IDT would generate eight 2D cross-sections of  $\mu_o$  in one design variable, or four 3D cross-sections of  $\mu_o$  in two design variables.

#### 4.1.1 Example: Development of a Turbofan Engine

A typical aircraft engine development program begins with the receipt of a Request for Proposal (RFP) from the aircraft engine user (the customer) [35]. The RFP, a requirements document describing the final flying characteristics of an aircraft to be developed, is the culmination of exploratory discussions between the customer and potential suppliers. Suppose that after examination of the RFP and discussions with airframe companies, the design team has decided that a particular turbofan configuration will be required. New aircraft engines are usually derived from existing designs [13]. In this example, the design team has two options:

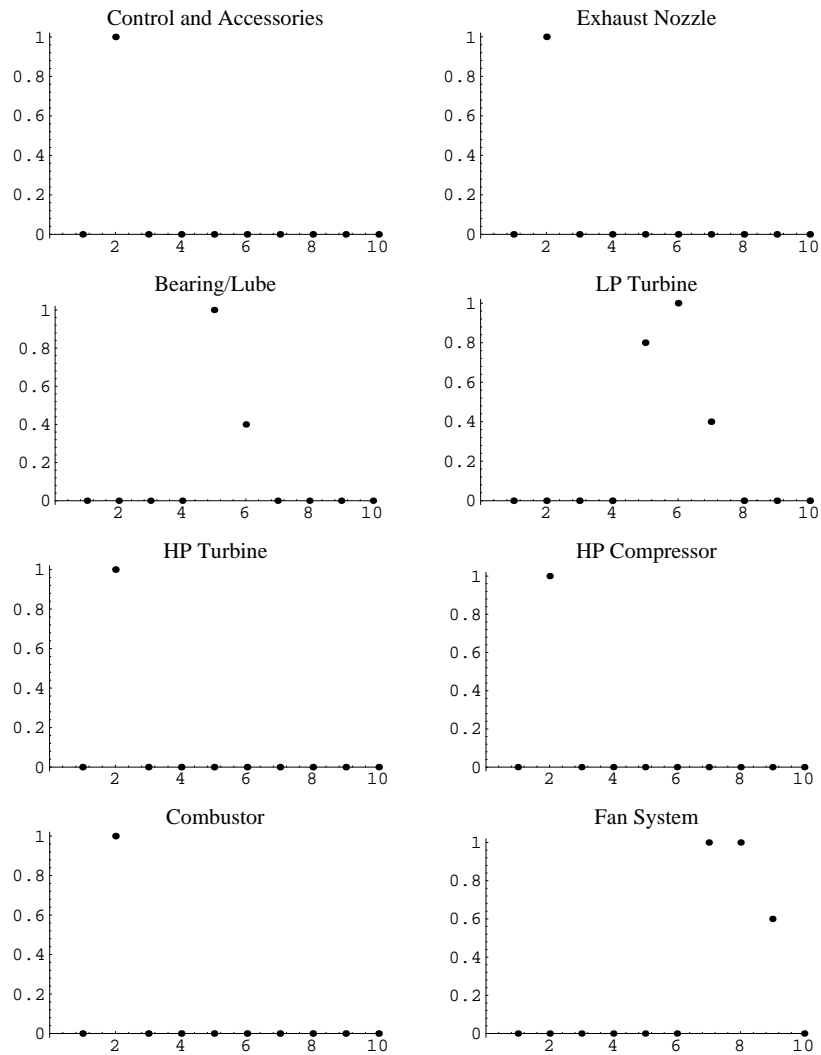
1. Develop the new engine from an existing turbojet design by the addition of a front fan with matching shaft and low pressure turbine (Figure 4.1). The turbojet engine will require minimal redesign to satisfy the RFP, but the addition of a fan, shaft, and low pressure turbine, even if taken from an existing engine, is a major design change.

2. Modify an existing, but dated, turbofan design. No major design changes will be necessary, but many subsystems will need to be modified.

At this preliminary stage of design, the design team must decide which option to pursue. A key consideration is the total development cost for the engine, which can be estimated by the EDCE. The degree of design change that the designers feel will be required in the eight subsystems of the new engine is imprecise, but the IDT permits the imprecise nature of this design information to be retained.

The method of imprecision begins with the specification of the individual design preferences  $\mu_{d_1}, \dots, \mu_{d_8}$  and the functional requirement  $\mu_p$ , and the identification of an appropriate hierarchy of design trade-offs. In order to correctly identify appropriate design trade-offs, it is first necessary to clearly distinguish what the various design preferences and functional requirement represent for this particular problem. The design variables  $d_1, \dots, d_8$  are inputs to the EDCE that represent the degree of innovation or design change in eight subsystems of the gas turbine engine to be developed. The single performance variable  $p = f(\vec{d})$  quantifies the development cost that corresponds to the “design”  $\vec{d}$ . Yet  $\vec{d}$  specifies not a particular engine design, but a particular set of degrees of design change, specified for each subsystem. Different points  $\vec{d} \in DVS$  represent, for example, the choice between a minor change to a different hole pattern in the combustor and a slightly more significant change to different nozzles.

The functional requirement  $\mu_p(p)$  represents the customer’s preferences among values of development cost  $p$ . These preferences on development cost are easily understood but less easily quantified. Clearly, lower development cost is preferable and this corresponds to a functional requirement with decreasing preference for increasing cost. Yet the exact variation of  $\mu_p(p)$  with  $p$  would, for a real design problem, depend on many factors and would hence be difficult to define. The functional requirement  $\mu_p(p)$  specified here decreases linearly from one, at the minimum development cost (for a turbofan engine with no design modifications), to zero, at the maximum development cost (see Figure 4.5 or 4.6). This choice of  $\mu_p$ , though reasonable, is arbitrary.



**Figure 4.3** Design preferences  $\mu_{d_1}, \dots, \mu_{d_8}$  for the turbojet option.

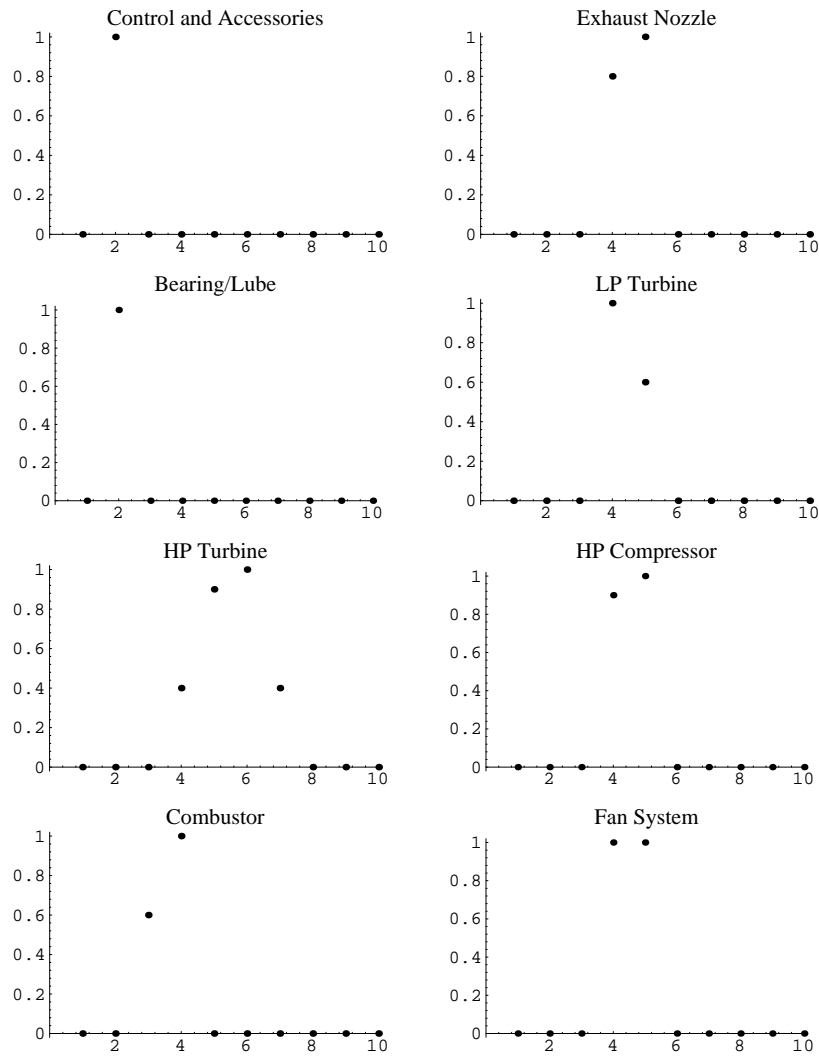
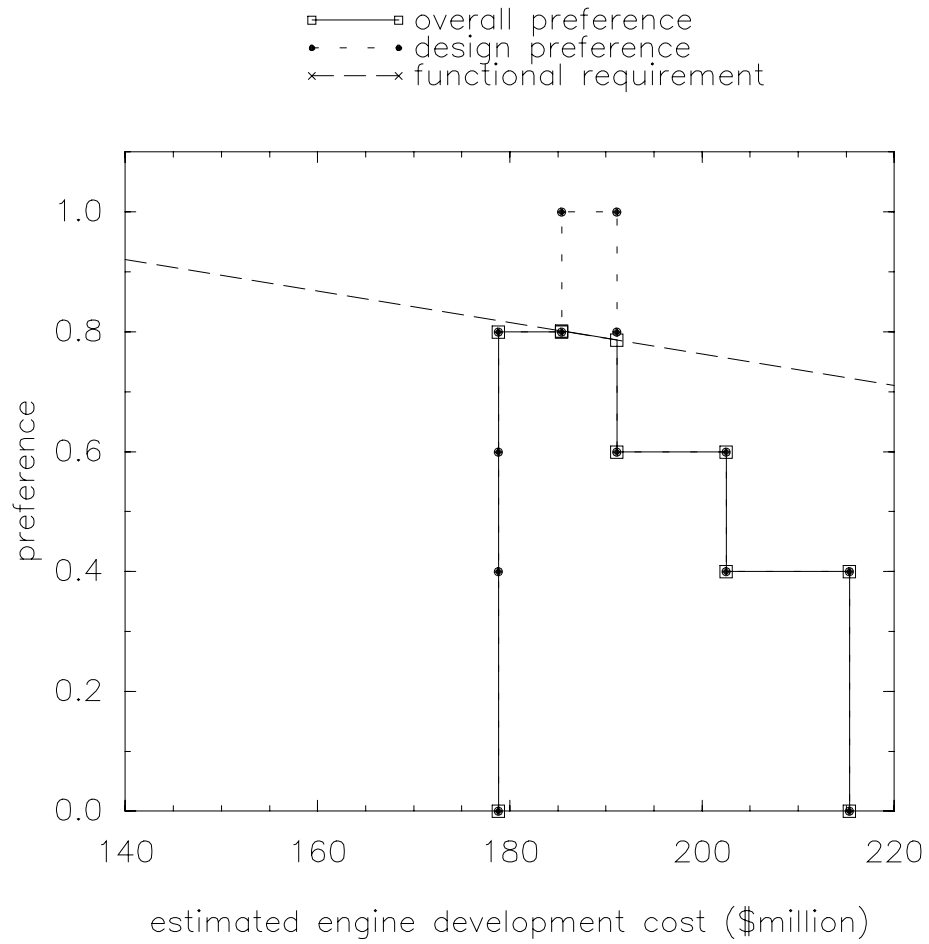
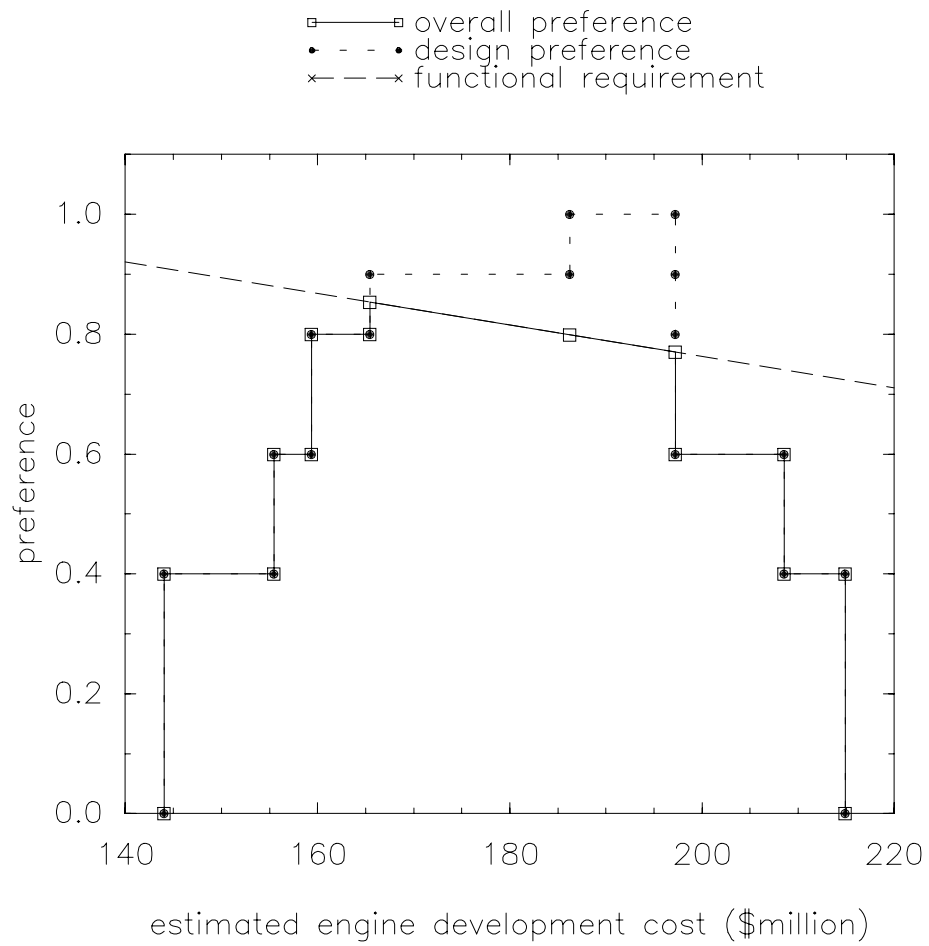


Figure 4.4 Design preferences  $\mu_{d_1}, \dots, \mu_{d_8}$  for the turbofan option.



**Figure 4.5**  $\mu_o$ ,  $\mu_d$ , and  $\mu_p$  on the PVS, for augmenting the turbojet.



**Figure 4.6**  $\mu_o$ ,  $\mu_d$ , and  $\mu_p$  on the PVS, for modifying the turbofan.

Recall that design preferences  $\mu_{d_i}$  represent the preference that the designers have for values of each design variable  $d_i$  based on aspects of design performance that are not already represented by performance variables. Development cost is already modeled as a performance variable. Given the high level, abstract nature of the design variables defined by the EDCE, specific measures of engine performance are not yet relevant. A more compelling consideration is the feasibility of a particular design  $\vec{d}$ : whether the levels of design change defined by  $\vec{d}$  will support an engine design that meets the specifications in the RFP. An excessively high level of design change, however, introduces unnecessary complexity into the engine design. Moreover, both of these considerations must be assessed separately for each design option. Thus feasibility and necessity of a particular level of design change is determined in the context of either augmenting the turbojet or modifying the turbofan. This leads to a different set of design preferences for each design option.

The design preferences determined by the design team based on these considerations are shown in Figures 4.3 and 4.4. Note that this example examines two separate design options that represent two markedly different types of engine development programs. It is necessary to deal with these two options separately, even though the same functional requirement on cost will be applied for both and even though the two different sets of design preferences specified over the same design variables  $d_1, \dots, d_n$  may overlap. The design preferences differ because feasibility is assessed relative to each engine development option. Thus the imprecise specification of each development program is realized in the design preferences shown in Figures 4.3 and 4.4.

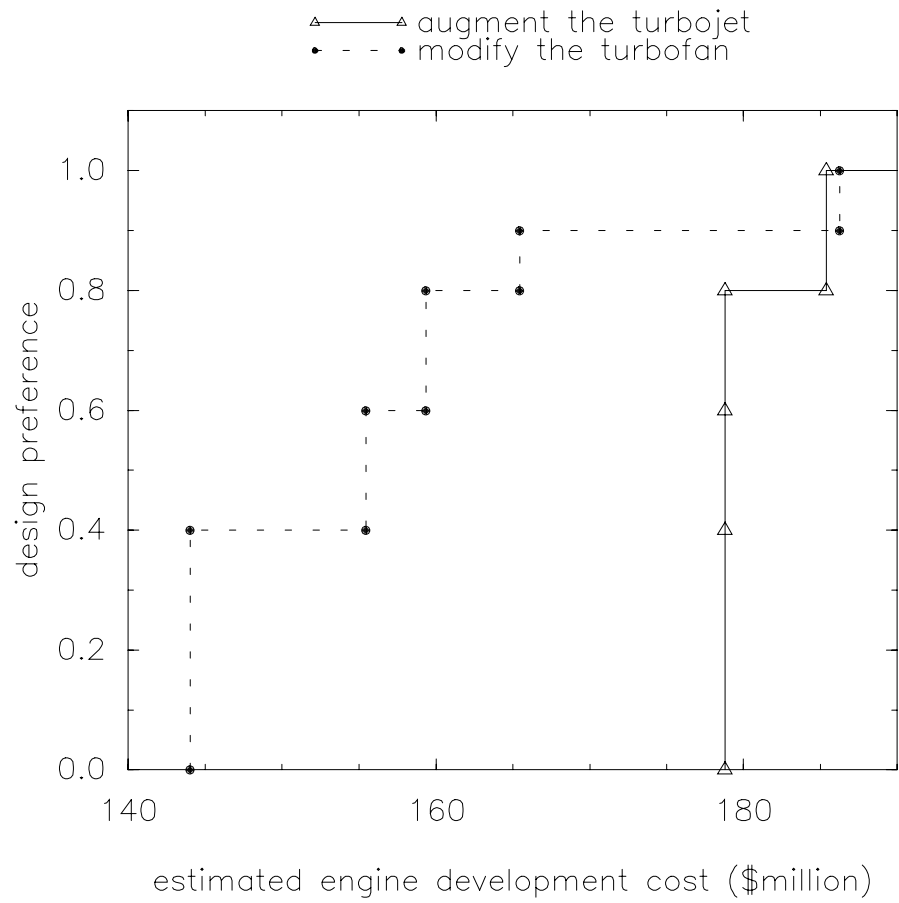
How then, should the design preferences be traded-off against each other? A highly feasible and necessary level of design change in one subsystem cannot compensate for a less feasible or excessive level of design change in another subsystem. Thus a non-compensating trade-off (aggregation function  $\mathcal{P}_{\min}$ ) is indicated for combining  $\mu_{d_1}, \dots, \mu_{d_8}$ . The correct trade-off between the combined design preference  $\mu_d$  and the functional requirement  $\mu_p$  is less obvious. To what extent does a highly feasible and necessary set of design change levels compensate for a high develop-

ment cost? Conversely, to what extent does a low development cost compensate for a less feasible or excessive set of design change levels? As a first approximation, a non-compensating trade-off is assumed. It will be shown below that the difficulty of quantifying the functional requirement on cost renders the exact trade-off between  $\mu_d$  and  $\mu_p$  moot.

Figures 4.5 and 4.6 show  $\mu_d(p)$  and  $\mu_o(p)$  calculated by the IDT for each of the two options. The development costs shown are representative and were not calculated using actual cost data. For augmenting the turbojet (option 1), the peak preference  $\mu_o^*$  is equal to 0.80 at an estimated development cost of \$185 million. For modifying the turbofan (option 2),  $\mu_o^*$  is equal to 0.85 at an estimated development cost of \$165 million. For the design preferences, functional requirement, and design strategy specified, the turbofan option results in a higher peak preference, suggesting that it is the better design choice.

Figures 4.5 and 4.6 show that the functional requirement  $\mu_p(p)$  truncates the combined design preference  $\mu_d(p)$  to produce the overall preference  $\mu_o(p)$ . This is because the non-compensating trade-off simply takes the min of  $\mu_d(p)$  and  $\mu_p(p)$ . As given,  $\mu_p(p)$  correctly reflects a relative preference for designs with lower cost  $p$ , but  $\mu_p(p)$  is an absolute measure. This is an important distinction because absolute preference on cost is traded-off against absolute preference on the feasibility and necessity of design changes and the min of the two is the measure by which the design is assessed. Figures 4.5 and 4.6 demonstrate that lowering  $\mu_p(p)$  results in lower peak overall preference  $\mu_o^*$ , corresponding to lower design preferences and a different set of peak preference designs. The correct decision therefore hinges on quantifying the customer's absolute preference on cost. But the customer's absolute preference for cost cannot be accurately quantified if it is no more specific than a desire to minimize cost.

Assuming that only the direction of the slope of  $\mu_p(p)$  is known, consider the effect of varying the point at which  $\mu_p(p)$  intersects the  $\mu_d(p)$  pyramid. On the  $\mu_d(p) = 1$  plateau  $[p_{\min}^1, p_{\max}^1]$ , the lowest cost point  $p_{\min}^1 = \$185$  million must be preferred over all other points in the interval. Indeed,  $p_{\min}^1$  must be preferred over



**Figure 4.7**  $\mu_d$  on the PVS for the turbojet and turbofan options.

all points to the right on the pyramid, since these points have cost  $p > p_{\min}^1$  and preference  $\mu(p) \leq 1$ . Hence only points on the left side of the pyramid where  $p \leq p_{\min}^1$  are of interest. The point at which  $\mu_p(p)$  intersects  $\mu_d(p)$  determines which performance  $p \leq p_{\min}^1$  achieves the peak overall preference  $\mu_o^*$ .

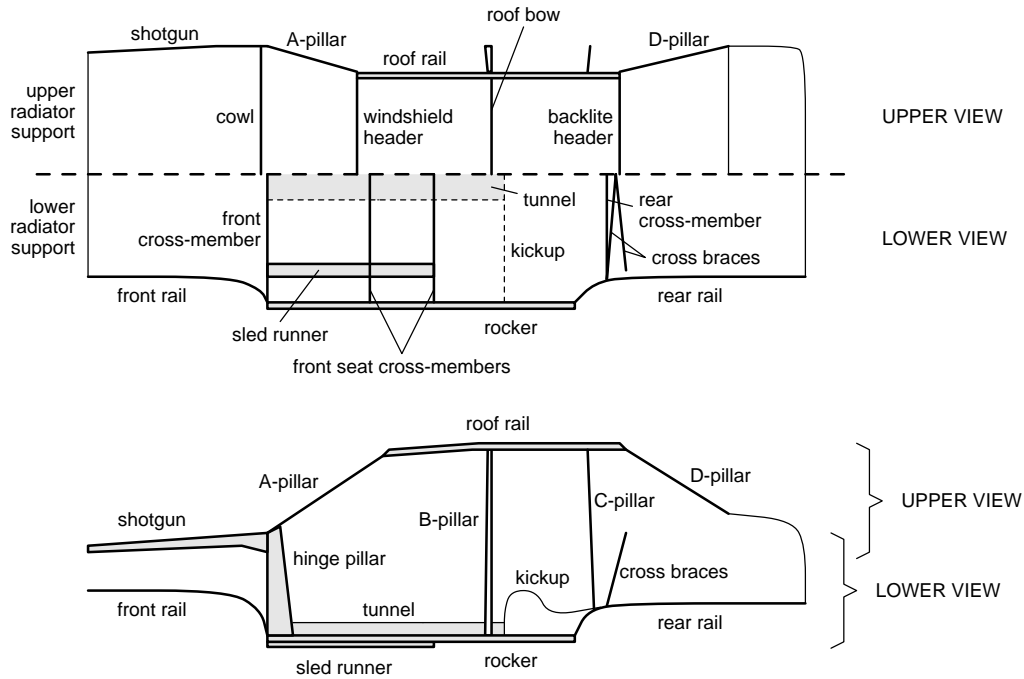
Instead of relying on a well-defined absolute functional requirement  $\mu_p$  to define the most preferred performance, a less formal method may be used. Figure 4.7 compares only the design preference  $\mu_d(p)$  for the two options. The customer's preferences on cost may now be informally applied to the quantified design preferences to identify preferred performances  $p \in \mathcal{Y}^*$ . In the absence of a sufficiently strong preference for minimizing cost, the most preferred performance is  $p_{\min}^1 = \$185$  million on the  $\mu_d(p) = 1$  plateau.  $p_{\min}^1$  can be considered to be a baseline point. As the desire to reduce development cost increases, points to the left of  $p_{\min}^1$  with lower development cost, but lower design preference, become increasingly attractive. This reflects an informal trade-off between lower development cost and lower feasibility implied by lower design preference. Even though this trade-off is not formally modeled as a specific aggregation operation on well defined preference functions  $\mathcal{P}(\mu_d, \mu_p)$ , it can still be represented on Figure 4.7. At  $\mu_d(p) = 0.9$ ,  $p_{\min}^{0.9} = \$165$  million for augmenting the turbojet is preferred over  $p_{\min}^{0.9} = \$185$  million for modifying the turbofan. Consider the choice between  $p_{\min}^{0.9} = \$165$  million for augmenting the turbojet and the baseline  $p_{\min}^1 = \$185$  million for modifying the turbofan. A straight line drawn through the two points is a simple indicator of the trade-off involved. A steeper slope yields a smaller decrease in cost for the same decrease in design preference. In the absence of any specific preferences on cost, such as a maximum budget of \$160 million, the overall preferred performance will either be the baseline point or the point that defines (or one of the co-linear points that together define) the shallowest trade-off line. Any point on the  $\mu_d$  pyramid below this line involves a steeper and presumably less desirable trade-off between development cost and design preference. For this example, the two points that the designers should choose between (assuming that no additional considerations enter into the decision) are  $p_{\min}^1 = \$185$  million for modifying the turbofan and  $p_{\min}^{0.9} = \$165$  million for augmenting the turbojet.

If some additional constraint rules out  $p_{\min}^1 = \$185$  million for the turbofan option, the next shallowest trade-off is given by  $p_{\min}^{0.8} = \$159$  million for the turbojet option. Thus a non-specific, relative preference on development cost can be informally applied to the combined design preference on performance  $\mu_d(p)$  in order to highlight promising performances from which a peak preference set of performances  $\mathcal{Y}^*$  can be chosen.

The results presented are from an early version of the IDT that used the original LIA. 12 function evaluations were required to map the design preferences  $\mu_{d_1}, \dots, \mu_{d_8}$  onto the PVS for the first alternative and 128 for the second. The design preferences for the two alternatives had 4 and 5  $\alpha$ -cuts with some coincident end points, especially for the first alternative, and the number of dimensions in the search space was 3 and 6. A more recent version of the IDT that implemented Powell's method modified for a discrete DVS reproduced the same results. Without taking advantage of monotonicity this later version of the IDT required 12 and 38 function evaluations for the two alternatives. As expected, optimization has a greater advantage for larger  $n$ . An example demonstrating the more advanced preference calculation techniques introduced in Chapter 3 is presented in Section 4.2.1.

## 4.2 Vehicle Structure Design

A more recent application of the IDT was motivated by discussions with engineers at a major U.S. automobile manufacturer. These engineers were primarily involved in design for noise, vibration, and harshness (NVH). NVH design is concerned with three aspects of vibration: audible noise, tactile vibration, and subjective evaluations of safety, comfort, and luxury based on perceived levels of noise and vibration. Measures of static rigidity (stiffness in bending and torsion) are indicators of perceived safety, comfort, and luxury levels: overly compliant vehicles rate poorly, though extremely stiff vehicles may suffer from high-frequency noise. Generally, different classes of vehicles and different body types have different standards for bending and torsional stiffness. Measures of dynamic response (modal frequencies



**Figure 4.8** The body-in-white.

and shapes, for the whole vehicle structure as well as for specific panels) directly predict noise and vibration characteristics [12].

The body-in-white is the principal load-bearing structure of the vehicle consisting of thin-walled parts welded, bolted, or glued together, including the windshield and backlite (rear window) which are structurally significant (Figure 4.8). Chassis and powertrain components, trim items (decklid, *i.e.*, trunk lid, dashboard, doors, hood), and components attached using a rubber mount are excluded. Although the parts excluded from the body-in-white add to the overall stiffness of the vehicle, they also add mass, and thus the static rigidity and dynamic response of the complete vehicle is considered to be well modeled by the body-in-white [12].

At this particular U.S. automobile manufacturer, the static and dynamic response of the body-in-white is calculated using a commercial finite element package, usually on a supercomputer. Two types of finite element model are used: detailed and simplified. A detailed model typically contains 30,000-100,000 elements and

requires 15 people about 2 months to construct by hand, using shape information from stylists. Detailed models are considered accurate: static stiffnesses are within 10% of test results [12].

A simplified model contains 5,000-10,000 elements. Simplified models were introduced at the company 2-3 years ago to reduce the calculation time required: they use fewer shell elements and approximate key structural members and joints with beam and spring elements. Preparing a simplified model from a detailed model requires about half the time needed to construct the detailed model, though sometimes a mix of parts from different models for different vehicles are used to create a new simplified model. The simplified model only approximates the detailed model: fewer elements correspond to fewer degrees of freedom and greater apparent stiffness. Thus the simplified model is 10-15% stiffer than the detailed model, and local behavior especially acoustic response is poorly modeled [12].

A key constraint was that each finite element calculation of static and dynamic response required significant supercomputer time: approximately 15 seconds to evaluate a simplified model [12]. Hence each function evaluation carried a significant cost, both directly in terms of billed supercomputer time and also in terms of the time delay. Therefore, in applying the method of imprecision to this problem, a key consideration was to minimize the number of function evaluations incurred. The optimization and experiment design techniques described in Chapter 3 were developed to address this issue. At this time, computational implementation of these methods is incomplete, and in particular, the IDT has not yet been extended to simultaneously map multiple performance variables.

Figure 4.9 shows the role of optimization and experiment design in the current version of the IDT. The information flow for one performance variable  $p_j = f_j(\vec{d})$  is indicated. The IDT begins by calling the experiment design module which conducts a central composite experiment over  $D_{\alpha_1}^d$ , the  $\alpha$ -cut with lowest  $\alpha$ , and constructs a linear approximation  $f'_j$ . Optimization is then called to search for the extrema  $p_{j_{\min}}^{\alpha_k}$  and  $p_{j_{\max}}^{\alpha_k}$  for each  $D_{\alpha_k}^d$ . Optimization uses the linear coefficients determined by the experiment design module to determine starting points. The linear approximation  $f'_j$

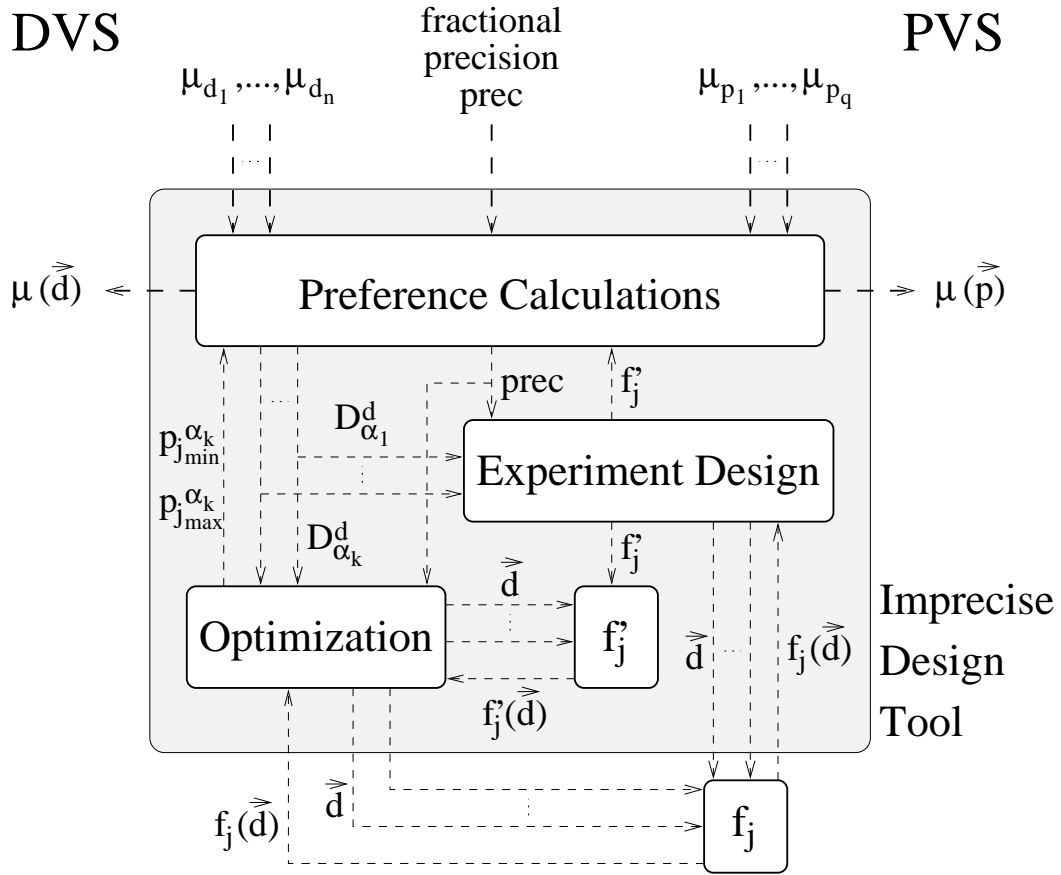
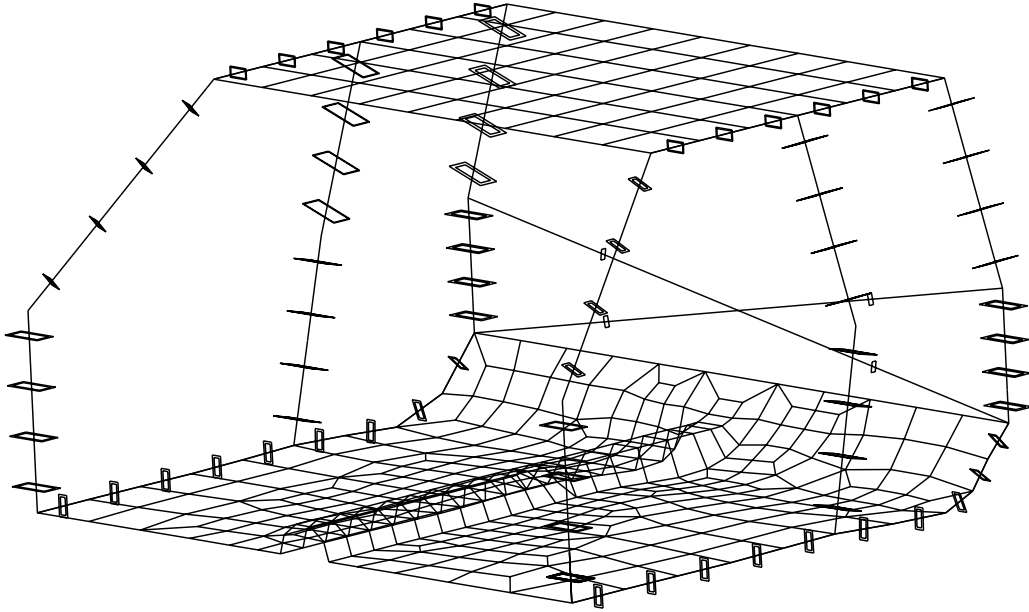


Figure 4.9 The Imprecise Design Tool.



**Figure 4.10** Finite element model of a car body.

replaces the function  $f_j$  for any design variables that are adequately approximated. The fractional precision “prec” used by the optimization and experiment design modules trades-off the number of function evaluations against accuracy.

#### **4.2.1 Example: Finite Element Analysis of a Vehicle Body**

The finite element model of the passenger compartment of an automobile body shown in Figure 4.10 was prepared by Michael Scott. It is not intended to be a realistic model of any particular vehicle or even of any real vehicle in general. The basic geometry, however, is similar to a typical four-door body-in-white as illustrated in Figure 4.8. In an effort to simulate part of the vehicle structure design process, a commercial finite element package was used to calculate bending and torsional stiffnesses using standard loads and support constraints obtained from [12].

This section discusses the application of the IDT to the problem of calculating two performance variables, bending and torsional stiffness, for ten design variables.

Bending stiffness  $K_B$  is defined as the total load applied at two pairs of matching points (*i.e.*, total of four) on the rocker rail, divided by the displacement averaged over several points on both sides of the vehicle body [12].  $K_B$  has units of pounds force per inch (lbf/in). Torsional stiffness  $K_T$  is defined as the torque applied between two pairs of matching load points, divided by the axial angle of twist measured between the pairs of load points [12].  $K_T$  has units of foot pounds force per degree (ft-lbf/°).

Ten design variables were selected (all units are inches, except where indicated otherwise):

$d_1$	0.10–0.20	B pillar gauge (thickness of hollow rectangular cross-section)
$d_2$	0.10–0.20	C pillar gauge
$d_3$	0.07–0.13	A pillar gauge
$d_4$	0.10–0.20	hinge pillar gauge
$d_5$	0.07–0.13	roof rail gauge
$d_6$	0.07–0.13	rocker gauge
$d_7$	0.03–0.05	floor gauge (plate thickness)
$d_8$	0.03–0.05	roof gauge
$d_9$	0.15–0.25	cross-sectional area of each cross-brace (square inches)
$d_{10}$	-2.0–2.0	fore-aft location of B pillar (fore is positive)

The ranges indicated are the maximum acceptable values assumed, which correspond to  $[d_{i\min}^\epsilon, d_{i\max}^\epsilon]$ .

Initially, the fractional precision was set to its highest value, 1, in order to test whether a minimally acceptable linear approximation could be constructed. Linear regression coefficients for bending stiffness calculated by the IDT for resolution III and resolution IV experiments are shown in Table 4.1. Design sensitivities  $\kappa_{1i}^\epsilon$  are shown for the resolution IV experiment.  $a_{1i}^\square$  denotes regression coefficients calculated from the fractional factorial experiment alone. Regression coefficients for the full central composite design are denoted  $a_{1i}$ . Note that apart from *B pillar*

design variable $d_i$	III		IV		$\kappa_{1i}^\epsilon$
	$a_{1i}^\square$	$a_{1i}$	$a_{1i}^\square$	$a_{1i}$	
B pillar gauge	14,400	13,800	10,800	10,800	0.053
C pillar gauge	21,700	22,300	25,100	25,100	0.123
A pillar gauge	29,600	30,300	33,600	33,700	0.099
hinge pillar gauge	26,400	26,800	28,500	28,500	0.139
roof rail gauge	153,000	155,000	160,000	161,000	0.471
rocker gauge	345,000	346,000	341,000	342,000	1.000
floor gauge	-17,600	18,700*	20,200	20,100	0.020
roof gauge	-23,900	4,390*	7,610	7,420	0.007
cross-brace area	-5,660	169*	-1,210	169*	0.001*
B pillar location	-340	-325	-207	-209	-0.041

**Table 4.1** Linear regression results for bending stiffness  $K_B$ .

*location*, all design variables should have positive coefficients for both  $K_B$  and  $K_T$ : thicker gauges should increase both bending and torsional stiffness. The negative resolution III coefficients  $a_{17}^\square, a_{18}^\square, a_{19}^\square$  for *floor gauge*, *roof gauge*, and *cross-brace area* in Table 4.1 imply that thickening the floor, roof, or cross-braces would decrease bending stiffness and are hence clearly invalid. The IDT flagged the corresponding  $a_{17}, a_{18}, a_{19}$  (originally negative) as not acceptably linear because they disagreed in sign with the observed slope in the design variables  $d_7, d_8$ , and  $d_9$  (all positive). In such a case the observed sign is more reliable than the calculated coefficient since relatively small non-linearities in a variable such as *rocker gauge* can strongly influence the calculated coefficient for less significant variables. Asterisks indicate that the coefficients for these not acceptably linear design variables are instead estimated from the one-factor-at-a-time experiment exclusively: they are therefore approximate. Note that the resolution III and IV experiments differ only in the fractional factorial: the one-factor-at-a-time experiment is identical.

The similarity between  $a_{1i}^\square$  and  $a_{1i}$  is potentially misleading:  $a_{1i}$  is obtained by adding the results from the one-factor-at-a-time experiment to  $a_{1i}^\square$ . Thus the agreement of the results for a particular resolution is to a certain extent guaranteed. Especially close agreement, however, for *hinge pillar gauge*, *roof rail gauge*, and

*rocker gauge* suggests that these variables are significantly linear. The largest approximation error at a point evaluated in a design variable direction for a linear approximation in the seven acceptably linear variables was 2,840 lbf/in (4%) at  $\vec{d}_6^-$  (*rocker gauge*). (The center point bending stiffness was 78,400 lbf/in, which is not unreasonable in comparison with actual vehicles [12].) Errors for *roof rail gauge* are also large, suggesting that  $K_B$  is significantly linear in these variables only relative to their large effect on  $K_B$ . The error in the approximation was similar for all seven acceptably linear variables. All of the errors were positive except at the center point, *i.e.*,  $\vec{f}^{\dagger} > \vec{f}$  at all evaluated points except  $\vec{d}_{\text{ctr}}$ . The error at the center point is equal to the offset  $\Delta_1$ , which was negative. The same was observed for the resolution IV experiment. This implies that both approximations overestimate bending stiffness near the exterior of  $D_e^d$  and underestimate bending stiffness near the center point.

It is assumed that the results from the resolution IV experiment are more reliable than the results from the resolution III experiment. The relatively close agreement of  $a_{1i}^{\square}$  and  $a_{1i}$  for resolution IV compared to resolution III is largely because of the increased number of points evaluated in the fractional factorial experiment (with center point): 13 versus 33. The number of additional points evaluated to complete the central composite design was 20 regardless of the resolution. For the approximation constructed in the nine acceptably linear variables, the largest approximation errors were measured for the fractional factorial corner points  $\vec{d}_1^{\square}, \dots, \vec{d}_{N^{\square}}^{\square}$ . The largest of these errors was 4,740 lbf/in (6%). Clearly, unmodeled non-linear effects exist. Errors were, however, largely symmetric, with  $E(\vec{d}_i^-) \approx E(\vec{d}_i^+)$ : no design variable was unevenly approximated. The limits chosen for the ten design variables were not narrow: bending stiffness varied from 56,900 lbf/in to 97,000 lbf/in.

For the resolution III experiment, the approximation constructed in the seven acceptably linear variables had a similar degree of accuracy. The largest error, for  $\vec{d}_2^{\square}$ , was 4,490 lbf/in (6%). Thus for the purposes of optimization,  $K_B$  is as well approximated by the resolution III experiment as by the resolution IV experiment, for those variables that were found to be acceptably linear for each resolution. But the resolution IV approximation linearizes nine variables compared to seven

design variable $d_i$	III		IV		$\kappa_{2i}^\epsilon$
	$a_{2i}^\square$	$a_{2i}$	$a_{2i}^\square$	$a_{2i}$	
B pillar gauge	19,500	19,900	20,600	20,700	1.000
C pillar gauge	1,760	2,100	4,100	4,110	0.198
A pillar gauge	9,040	8,930	8,350	8,340	0.242
hinge pillar gauge	-217	550*	962	938	0.045
roof rail gauge	17,500	17,400	15,900	15,900	0.461
rocker gauge	31,400	31,800	31,800	31,800	0.922
floor gauge	62,600	63,700	67,900	68,000	0.657
roof gauge	-13,500	4,570*	5,080	5,050	0.049
cross-brace area	-2,150	132*	81.7	84.6	0.004
B pillar location	-61.3	-58.0	-24.8	-25.6	-0.049

**Table 4.2** Linear regression results for torsional stiffness  $K_T$ .

variables for the resolution III approximation.

The design sensitivities  $\kappa_{1i}^\epsilon$  tabulated for the resolution IV experiment show that *rocker gauge* is the critical design variable. The estimated change in bending stiffness between the acceptable extremes of *roof rail gauge* ( $\kappa_{15}^\epsilon = 0.471$ ) is less than half the corresponding estimated change for *rocker gauge*. Bending stiffness also depends significantly on *hinge pillar gauge* ( $\kappa_{14}^\epsilon = 0.139$ ) and *C pillar gauge* ( $\kappa_{12}^\epsilon = 0.123$ ). Down the list it is seen that *roof gauge*, *floor gauge*, and *cross-brace area* are relatively unimportant variables in determining bending stiffness. Forces are largely being transmitted through the roof rail and rocker.

Linear regression coefficients and design sensitivities for torsional stiffness are shown in Table 4.2. For the resolution III experiment, three variables were invalid because the sign of the regression coefficient disagreed with the sign of the observed slope. For the resolution IV experiment, however, all ten variables were acceptably linear. The results for both bending and torsional stiffness demonstrate that a resolution IV experiment is likely to resolve more linear variables than a resolution III experiment. Additionally, Tables 4.1 and 4.2 suggest that variables with small design sensitivities tend to be less easily resolved. The largest approximation error for the resolution III approximation in seven variables was  $E(\bar{d}_9^\square) = 1,010$  ft-lbf/°

	III		IV	
	$\Delta_j$	$\Delta_j^{\mathcal{L}}$	$\Delta_j$	$\Delta_j^{\mathcal{L}}$
bending (lbf/in)	-1,230	-548	-1,240	-456
torsion (ft-lbf/°)	-261	-115	-258	-258

**Table 4.3** Center point offsets for bending and torsional stiffness.

(7%). (The center point torsional stiffness was 13,300 ft-lbf/°, which is comparable with actual vehicles [12].) The largest approximation error for the resolution IV approximation in all ten variables was  $E(\vec{d}_4^{\square}) = 1,350$  ft-lbf/° (11%). Torsional stiffness was overestimated near the exterior of  $D_e^d$  and underestimated near the center point. Errors were largely symmetric with  $E(\vec{d}_i^-) \approx E(\vec{d}_i^+)$ . These effects, which were common to  $K_B$  and  $K_T$ , may be an artifact of the finite element analysis.

The design sensitivities in Table 4.2 show that the two most important variables for torsional stiffness are *B pillar gauge* ( $\kappa_{21}^{\epsilon} = 1$ ) and *rocker gauge* ( $\kappa_{26}^{\epsilon} = 0.922$ ). The distinction is not as pronounced as for bending stiffness: torsional stiffness depends significantly on *floor gauge* ( $\kappa_{27}^{\epsilon} = 0.657$ ), *roof rail gauge* ( $\kappa_{25}^{\epsilon} = 0.461$ ), *A pillar gauge* ( $\kappa_{23}^{\epsilon} = 0.242$ ), and *C pillar gauge* ( $\kappa_{22}^{\epsilon} = 0.198$ ). Recall that *roof gauge*, *floor gauge*, and *cross-brace area* had low values of  $\kappa_{1i}^{\epsilon}$  for bending stiffness. Therefore, increasing *floor gauge* selectively increases torsional stiffness. To a lesser extent, increasing *hinge pillar gauge* selectively increases bending stiffness.

The offsets  $\Delta_j$  (for all ten variables) and  $\Delta_j^{\mathcal{L}}$  (for acceptably linear variables) are shown in Table 4.3. The offsets calculated from the fractional factorial points were not significantly changed when the one-factor-at-a-time experiment results were added and are thus not tabulated. This indicates that the quadratic error estimates from the two separate sets of points are similar. The agreement between the values for  $\Delta_j$  for the resolution III and IV experiments is also good. Comparisons involving  $\Delta_j^{\mathcal{L}}$  are not meaningful because the number of linear variables approximated differs. The offset determines the error at the center point.  $\Delta_1 = -1,240$  lbf/in is a -2% error.  $\Delta_2 = -258$  ft-lbf/° is also a -2% error. The error at the center point should

design variable $d_i$	$d_{i_{\min}}^\epsilon$	$d_{i_{\max}}^\epsilon$	prec = 1		prec = 0.01
			$p_{j_{\min}}^\epsilon$	$p_{j_{\max}}^\epsilon$	$p_{j_{\max}}^\epsilon$
B pillar gauge	0.10	0.20	0.10	0.20	0.20
C pillar gauge	0.10	0.20	0.10	0.20	0.20
A pillar gauge	0.07	0.13	0.07	0.13	0.13
hinge pillar gauge	0.10	0.20	0.10	0.20	0.20
roof rail gauge	0.07	0.13	0.07	0.13	0.13
rocker gauge	0.07	0.13	0.07	0.13	0.13
floor gauge	0.03	0.05	0.03	0.05	0.05
roof gauge	0.03	0.05	0.03	0.05	0.05
cross-brace area	0.15	0.25	0.15	0.25	0.25
B pillar location	-2.0	2.0	2.0	-2.0	-1.3
bending stiffness $K_B$			56,900	96,900	97,000
torsional stiffness $K_T$			9,400	16,900	16,900

**Table 4.4** Extrema for bending and torsional stiffness in  $D_\epsilon^d$ .

be smaller in magnitude than the error near the boundaries of  $D_\epsilon^d$  if it is indeed quadratic, as can be seen in Figure 3.12.

After the construction of a linear approximation, the IDT proceeded with optimization over the reduced set of non-linear design variables. A few additional function evaluations are used to verify the results of the optimization, which begins at the corner of the search space where the extremum is expected. The calculated extrema in bending and torsional stiffness achievable by designs  $\vec{d} \in D_\epsilon^d$  are shown in Table 4.4. For this example the extrema in bending and torsional stiffness found for both resolutions using a fractional precision of 1 occurred at the same design  $\vec{d}$ . Hence the results for prec = 1 are shown together. Extrema are not expected to coincide, in general. Despite the differences in accuracy of the linear approximations constructed, the same extrema were found for the two resolutions. There is, however, an issue of reliability and confidence: the approximation obtained through the resolution IV experiment is more complete and more points were evaluated to verify its accuracy. The additional function evaluations buy a degree of confidence in the validity of the results even though they are numerically identical.

		prec = 1		prec = 0.01	
		III	IV	III	IV
bending	approximation	41	53	33	53
	optimization	2	2	58	58
	total	43	55	91	111
torsion	approximation	41	53	33	53
	optimization	2	2	37	37
	total	43	55	70	90

**Table 4.5** Number of function evaluations required by the IDT.

To verify the extrema found, the fractional precision was reduced to 0.01. This tightened the criteria for acceptable linearity in a design variable such that all design variables became unacceptable. Optimization searched for extrema without any of the enhancements described in Chapter 3 that seek to minimize the number of function evaluations required. The results were again independent of the resolution specified. The extrema returned for torsional stiffness were identical. A new maximum bending stiffness was located at an intermediate *B pillar location* of 1.3 inches aft. The increase in  $K_B$  was tiny: 55 lbf/in (the new maximum bending stiffness was 97,000 lbf/in). The torsional stiffness at this new maximum of bending stiffness is shown in italics.

An important consideration is the number of function evaluations required to obtain a given reliability or accuracy in the calculated results. The number of function evaluations for the four combinations of resolution and fractional precision evaluated are shown in Table 4.5. The number of function evaluations required for ten variable resolution III and IV central composite designs is 33 and 53 (from Figure 3.11). The additional eight points evaluated in constructing a resolution III approximation for  $K_B$  and  $K_T$  were required to verify the seven variable approximations. These verifying points, which define the eight corners of a cube in the three non-linear variables, did not coincide with any of the 33 points evaluated in the resolution III central composite design. Table 4.5 shows that only two additional evaluations

were required by optimization when the fractional precision is 1, regardless of resolution, for both  $K_B$  and  $K_T$ . This implies not that optimization only examined two points, but that the linear approximation  $f'_j$  obviated the need to evaluate all but two points. These two points correspond to the extrema found. This demonstrates that the linear approximation methods described in Chapter 3 and implemented in the IDT are effective at reducing the number of additional evaluations required by optimization.

The number of optimization evaluations required for a fractional precision of 0.01 do not reflect the number of evaluations required by optimization alone. Although optimization rejects the linear approximation and proceeds on all ten design variables, it begins at the extrema predicted by the approximation. For torsional stiffness, optimization expends 37 function evaluations in order to determine that its starting points are indeed the correct extrema, to a fractional precision of 0.01. For bending stiffness, optimization expends 58 evaluations in order to verify the minimum and to find a maximum that differs only in *B pillar location*. Table 4.5 indicates the trade-off between fractional precision and the number of function evaluations, for this particular example. There is no simple relationship between the two. As the fractional precision is reduced, the IDT becomes more selective in approximating  $\vec{f}$ , and more cautious in searching for extrema, with smaller step sizes and more stringent termination criteria. This caution demonstrably results in more function evaluations, but it is only suggested without proof that it results in better results.

It is difficult to assess the quality of the results when they are numerically virtually identical. A difference in *B pillar location* of 0.7 inches resulting in a marginal increase in bending stiffness is almost negligible. The apparent effectiveness of the methods developed poses an intriguing question: if it is possible to obtain the correct answer with a resolution III experiment and a fractional precision of 1 requiring only 44 function evaluations, why would additional function evaluations be necessary? It has already been discussed that a resolution IV experiment buys additional reliability and confidence in the results, and a smaller fractional precision limits the

allowable inaccuracy. Yet this example appears to show that the central issue is not accuracy but confidence in the results, which is not easily assessed.

#### 4.2.2 Example: Quadratic Approximations to the Finite Element Analysis

An example involving closed-form equations for  $\vec{f}$  will further demonstrate the computational method. For this example, quadratic approximations in the two most poorly behaved design variables were constructed for bending and torsional stiffness. These quadratic approximations were then used to calculate  $K_B$  and  $K_T$  instead of the finite element package. The two design variables selected were *cross-brace area*  $d_9$  and *B pillar location*  $d_{10}$ . The approximations were constructed relative to the center point, in terms of  $x_9$  and  $x_{10}$ . The acceptable intervals at  $\alpha = \epsilon$  specified for  $x_9$  and  $x_{10}$  were equivalent to those specified in the previous example for *cross-brace area* and *B pillar location*. The quadratic approximation for bending stiffness is plotted over  $D_\epsilon^d$  in Figure 4.11:

$$(4.1) \quad K_B = 78,400 + 170x_9 - 240x_{10} - 630x_9^2 - 5x_9x_{10} - 88x_{10}^2$$

$K_B$  is significantly non-linear in *B pillar location*.  $\frac{\partial K_B}{\partial x_{10}} = 0$  on a line from (-0.05,-1.36) to (0.05,-1.36). This line defines a ridge on which  $K_B$  is a maximum for a given value of *cross-brace area*. This ridge cannot be modeled by a linear approximation. The true maximum of  $K_B$  on  $D_\epsilon^d$  (78,600 lbf/in) is located at (0.05,-1.36). The minimum (77,600 lbf/in) is located at (-0.05,2).

The quadratic torsional stiffness is plotted in Figure 4.12:

$$(4.2) \quad K_T = 13,300 + 130x_9 - 38x_{10} - 620x_9^2 + 5x_9x_{10} + 4x_{10}^2$$

The partial derivatives of  $K_T$  remain non-zero within the search space. The minimum (13,230 ft lbf/°) is at (-0.05,2). The maximum (13,400 ft lbf/°) is at (0.05,-2).

The IDT was used to search for extrema at two fractional precisions: 1 and 0.01. For two design variables, the full factorial experiment must be evaluated (all

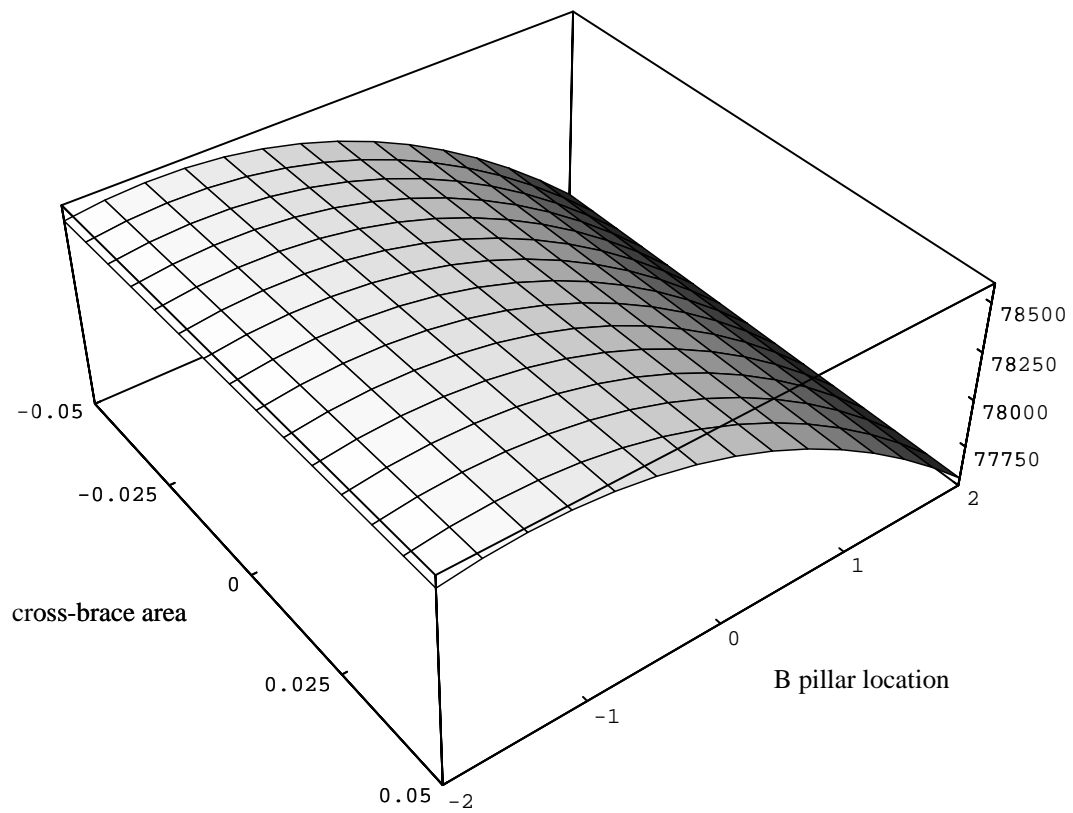


Figure 4.11 Quadratic bending stiffness.

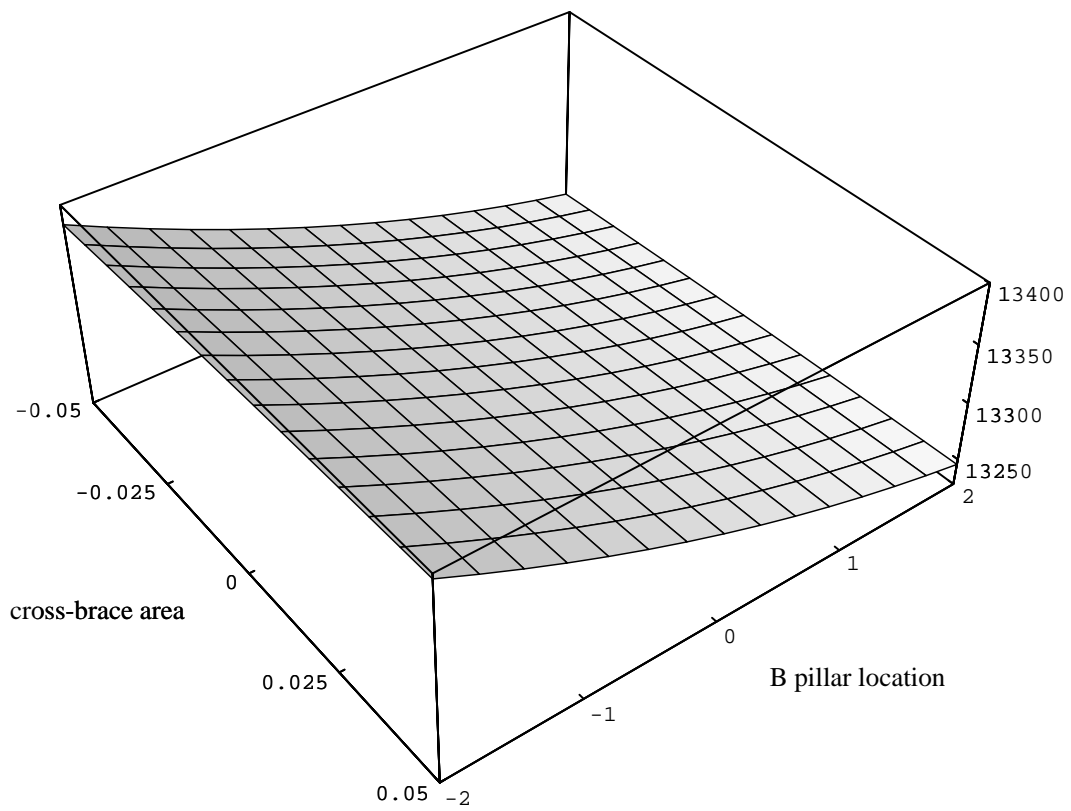


Figure 4.12 Quadratic torsional stiffness.

	$K_B$	$K_T$
cross-brace area $a_{j9}$	170	130
B pillar location $a_{j10}$	-240	-38
$\Delta_j$	-118	4.8

**Table 4.6** Linear regression results for quadratic  $K_B$  and  $K_T$ .

	prec = 1		prec = 0.01
	$p_{j\min}^\epsilon$	$p_{j\max}^\epsilon$	$p_{j\max}^\epsilon$
cross-brace area	-0.050	0.050	0.050
B pillar location	2.0	-2.0	-1.39
$K_B$	77,600	78,500	78,600
$K_T$	13,230	13,400	(13,370)

**Table 4.7** Extrema for quadratic  $K_B$  and  $K_T$ .

four corners of the  $D_\epsilon^d$  rectangle): there is no choice between resolution III and resolution IV experiments. The linear approximation results for  $K_B$  and  $K_T$  are given in Table 4.6. The correct linear regression coefficients from Equations (4.1) and (4.2) were recovered. The offset  $\Delta_1$  is relatively large and negative in order to correct for the concave downwards curvature of  $K_B$ .  $\Delta_2$  is smaller in comparison and positive:  $K_T$  is not strongly non-linear and is concave upwards.

The extrema found by the IDT are shown in Table 4.7. The extrema found using the linear approximation at a fractional precision of 1 coincided for  $K_B$  and  $K_T$ . At a fractional precision of 0.01, optimization identified a maximum for  $K_B$  at (0.05,-1.39). The value for  $K_B$  at (0.05,-1.39) was identical, within the floating point precision, to the true maximum value of  $K_B$ . At a fractional precision of 0.01, only one variable,  $x_{10}$ , was rejected as not acceptably linear, for both  $K_B$  and  $K_T$ . Errors for the linear approximations in both variables (prec = 1) and in one variable (prec = 0.01) are given in Table 4.8 for each evaluated point. Evaluated points are

	$x_9$	$x_{10}$	$K_B$	$K_T$	$E(\vec{d})$			
					prec = 1		prec = 0.01	
					$K_B$	$K_T$	$K_B$	$K_T$
$\vec{d}_{\text{ctr}}$	0	0	78,400	13,300	-118	4.8	-0.5	-0.5
$\vec{d}_1^{\square}$	-0.05	-2	78,500	13,380	236	-10.2	1.6	0.5
$\vec{d}_2^{\square}$	-0.05	2	77,600	13,230	235	-9.2	0.6	1.5
$\vec{d}_3^{\square}$	0.05	-2	78,500	13,400	235	-9.2	0.6	1.5
$\vec{d}_4^{\square}$	0.05	2	77,600	13,250	236	-10.2	1.6	0.5
$\vec{d}_9^-$	-0.05	0	78,400	13,290	-116	6.3	1.1	1.0
$\vec{d}_9^+$	0.05	0	78,400	13,310	-116	6.3	1.1	1.0
$\vec{d}_{10}^-$	0	-2	78,500	13,390	234	-11.2		
$\vec{d}_{10}^+$	0	2	77,600	13,240	234	-11.2		

**Table 4.8** Approximation errors for quadratic  $K_B$  and  $K_T$ .

shown in Figure 4.13. Although  $E(\vec{d}_4^{\square}) = 236$  lbf/in for  $K_B$  is only a 0.3% error, the fractional precision is applied to the observed range in  $K_B$  (77,600–78,500 lbf/in). Hence for a fractional precision of 0.01, the maximum allowable approximation error in  $K_B$  is  $0.01 \times 900 = 9$  lbf/in. Similarly  $E(\vec{d}_{10}^-) = -11.2$  ft lbf/ $^{\circ}$  for  $K_T$  is only a 0.08% error, but  $0.01 \times 170 = 1.7$  ft lbf/ $^{\circ}$ .

Figure 4.13 shows  $D_{\epsilon}^d$ , the design preference  $\alpha$ -cut in the DVS. The factorial experiment points are numbered. The actual and approximate mappings of  $D_{\epsilon}^d$  onto the PVS are shown in Figure 4.14. The solid lines indicate the boundary of  $P_{\epsilon}^d$ . The pronounced non-linearity of  $K_B$  in  $x_{10}$  results in a boundary that is not only curved, but also crosses over itself. The maximum in  $K_B$  occurs at (0.05,-1.36). The dashed lines connecting the numbered factorial experiment points indicate the boundary of  $P_{\epsilon}^{d'}$ , obtained via linear approximation. For this example, which was chosen to highlight the two most non-linear design variables,  $P_{\epsilon}^{d'}$  is a poor approximation to  $P_{\epsilon}^d$ , especially near  $\vec{d}_1^{\square}$  and  $\vec{d}_3^{\square}$ . The mismatch between  $P_{\epsilon}^{d'}$  and  $P_{\epsilon}^d$  could be detected by checking if  $\vec{p}(\vec{d}_{\text{ctr}})$ , the performance of the center point, is inside  $P_{\epsilon}^{d'}$ . Looking at Figure 4.14, it is tempting to suggest that  $P_{\epsilon}^d$  could be estimated by connecting the dots, but this is difficult to generalize to  $n$  design variables. Using

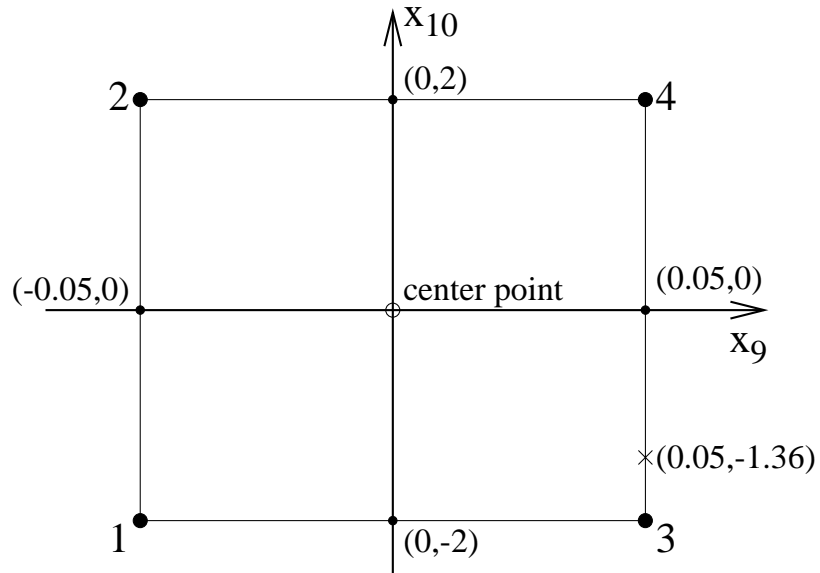


Figure 4.13 The  $\alpha$ -cut  $D_\epsilon^d$ .

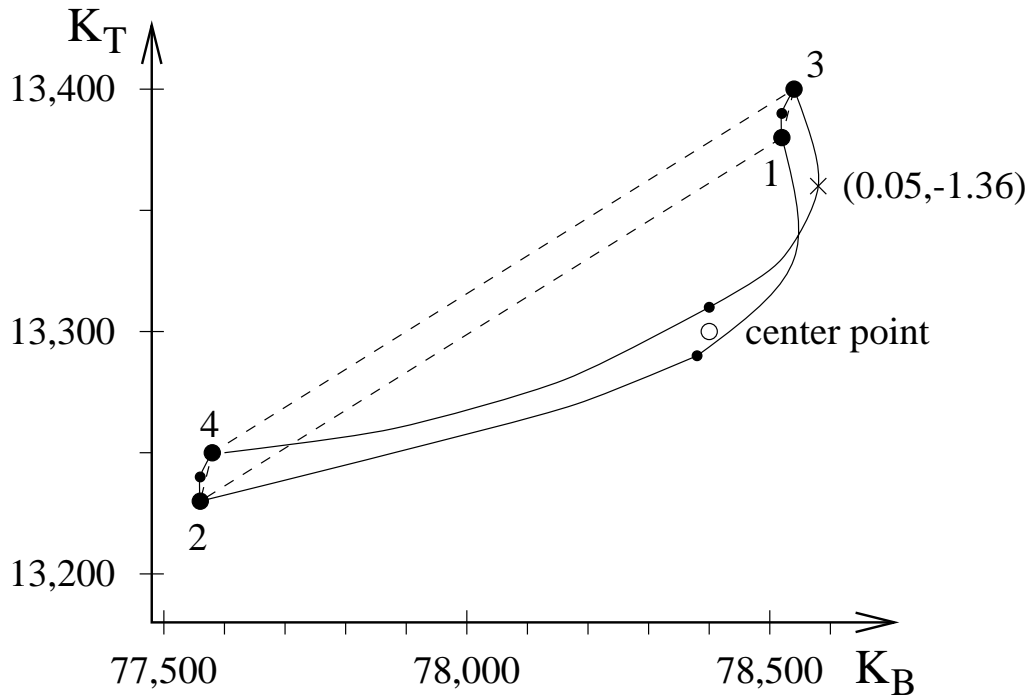


Figure 4.14  $D_\epsilon^d$  mapped onto  $P_\epsilon^d$  and  $P_\epsilon^{d'}$ .

$P_\epsilon^{d\Box} = [p_{1\min}^\epsilon, p_{1\max}^\epsilon] \times [p_{2\min}^\epsilon, p_{2\max}^\epsilon]$  would require no additional function evaluations but would grossly overestimate  $P_\epsilon^d$ .

Where  $\vec{f}$  is sufficiently non-linear that  $P_\epsilon^{d'}$  obtained via linear approximation is inadequate, effective methods to more accurately determine the geometry of  $P_\epsilon^d$  have yet to be developed within the method of imprecision. Nevertheless, this example shows that even where variables are non-linear, the correct extrema can be located and linear approximation can still be valuable in facilitating optimization.

### 4.3 Conclusions

The Imprecise Design Tool (IDT) was developed to verify the algorithms described in Chapter 3 and to demonstrate the method of imprecision on engineering problems from industry. Section 4.1 discussed an application of the IDT to the Engine Development Cost Estimator (EDCE) provided by General Electric Aircraft Engines, Cincinnati, Ohio. The EDCE estimates the cost of developing a new aircraft engine and is one of several programs that together estimate the total lifetime cost of an engine. Eight variables that represent the degree of innovation in eight components and subsystems of the new engine were chosen to be design variables  $d_1, \dots, d_8$ . These variables are significantly imprecise: the degree of innovation for a particular component of the engine will, in general, not be known precisely in advance. Additionally, because the EDCE defines degree of innovation only at ten levels,  $d_1, \dots, d_8$  are effectively discrete.

The EDCE calculates the estimated development cost for a new engine given a set of precisely specified, crisp inputs. The role of the IDT was to provide an interface that quantified the imprecision in the design variables  $d_1, \dots, d_8$  and the performance variable  $p$ . The EDCE defined the crisp mapping  $f : DVS \rightarrow PVS$  used by the IDT to perform preference calculations.

This early version of the IDT traded-off preferences using the non-compensating aggregation function  $\mathcal{P}_{\min}$ . The resulting  $n$ -cubic combined design preference  $\alpha$ -cuts  $D_{\alpha_k}^d$  were mapped onto the PVS using a modified form of the LIA that made use of

a lookup table for values of  $f(\vec{d})$ . Calculations demonstrated that specifying  $\alpha$ -cuts at ten levels of preference was computationally inefficient and unnecessary.

An early solution to the problem of mapping  $\mathcal{Y}^*$  back onto the DVS to obtain  $\mathcal{X}^*$  used the  $\alpha$ -cut at  $\alpha^*$ , the largest  $\alpha_k \leq \mu_o^*$ . For a non-compensating trade-off, the set of peak preference designs  $\mathcal{X}^*$  is a subset of  $D_{\alpha^*}^d$  because only design configurations  $\vec{d} \in D_{\alpha^*}^d$  can have overall preference  $\mu_o(\vec{d})$  equal to  $\mu_o^*$ .

Section 4.1.1 presented a turbofan aircraft engine development problem which involved two (imprecisely specified) options:

1. Develop the new engine from an existing turbojet design by the addition of a front fan with matching shaft and low pressure turbine.
2. Modify an existing, but dated, turbofan design.

The IDT was used to map design preferences onto the one-dimensional PVS. The specification of a relative functional requirement for minimizing cost was shown to be problematic. The usual calculation of overall preference  $\mu_o(p)$  by aggregating  $\mu_d(p)$  and  $\mu_p(p)$  requires  $\mu_p(p)$  to be defined as an absolute functional requirement. Instead, an informal method for representing a purely relative preference on cost by comparing the trade-offs implied by particular points on the  $\mu_d(p)$  pyramid was described.

A later version of the IDT that replaced the LIA with optimization demonstrated the application of optimization to a design problem with discrete design variables. It was also shown that optimization was able to reduce the number of function evaluations, in one particular case, from 128 to 38. A simpler calculation required 12 evaluations for both methods.

Section 4.2 presented a more recent application of the IDT, to an automobile structure design problem. Noise, vibration, and harshness (NVH) design is concerned with three aspects of vibration: audible noise, tactile vibration, and subjective evaluations of safety, comfort, and luxury based on perceived levels of noise and vibration. Measures of static rigidity (bending and torsional stiffness) are indicators of perceived safety, comfort, and luxury levels. Measures of dynamic response

(modal frequencies and shapes) directly predict noise and vibration characteristics [12].

The body-in-white is the principal load-bearing structure of the vehicle consisting of thin-walled parts welded, bolted, or glued together, including the windshield and rear window. At this particular U.S. automobile manufacturer, the static and dynamic response of the body-in-white is calculated using a commercial finite element package. Two types of finite element model are used: detailed and simplified. Simplified models reduce calculation time, yet each finite element calculation of static and dynamic response still requires approximately 15 seconds [12]. Therefore, in applying the method of imprecision to this problem, a key consideration was to minimize the number of function evaluations incurred. The optimization and experiment design techniques described in Chapter 3 were developed to address this issue.

In Section 4.2.1 the IDT was applied to a finite element model of the passenger compartment of a hypothetical four-door body-in-white. A commercial finite element analysis package was used to calculate bending and torsional stiffnesses. Ten design variables were defined including the gauges of key members and the fore-aft location of the B pillar. The example not only illustrates the methods introduced in Chapter 3, but also attempts to demonstrate their feasibility. The use of finite element models is not limited to automobile structure design: they are widely used in industry. Detailed results for resolution III and IV central composite experiment designs were presented. The largest known error in approximating bending stiffness  $K_B$  (acceptably linear variables only) was 6% for both resolutions. For  $K_B$ , the resolution III experiment identified seven acceptably linear variables and the resolution IV experiment identified nine acceptably linear variables. The largest known error in approximating torsional stiffness  $K_T$  was 7% for resolution III (seven variables) and 11% for resolution IV (all ten variables). Both  $K_B$  and  $K_T$  were overestimated near the exterior of  $D_c^d$  and underestimated near the center point. Errors were largely symmetric with  $E(\vec{d}_i^-) \approx E(\vec{d}_i^+)$ . These errors suggest that all four linear approximations ( $K_B$ , resolutions III and IV, and  $K_T$ , resolutions III and IV)

were reasonably accurate. Furthermore, the offsets  $\Delta_j$  appeared to be effective in evening out errors such that no subset of the search space was especially poorly approximated.

Calculated design sensitivities showed that for the  $\alpha = \epsilon$  design preference intervals specified, bending stiffness was most sensitive to the design variable *rocker gauge*. Torsional stiffness was most sensitive to the design variables *B pillar gauge* ( $\kappa_{21}^\epsilon = 1$ ) and *rocker gauge* ( $\kappa_{26}^\epsilon = 0.922$ ).

Optimization at fractional precisions of 1 and 0.01 found the same minima for  $K_B$  and  $K_T$  on  $D_\epsilon^d$  using experiment design information at both resolutions. The minima for  $K_B$  and  $K_T$  coincidentally occurred at the same point in the DVS. Optimization at a fractional precision of 1 found the same maxima for  $K_B$  and  $K_T$  on  $D_\epsilon^d$  for both resolutions. These maxima also occurred at a single point in the DVS. A maximum with a negligibly higher value of  $K_B$  was found when the fractional precision was set to 0.01.

Comparing the number of function evaluations required for resolution III and IV experiments and fractional precisions of 1 and 0.01 showed that, for this example, the results and the number of function evaluations required for optimization were independent of resolution. Approximation was also shown to be effective in minimizing the number of additional function evaluations required by optimization: for a fractional precision of 1, only two additional function evaluations were required. Moreover, even a resolution III experiment combined with a fractional precision of 1 resulted in calculated extrema that were extremely close to the best extrema found. The apparent effectiveness of the methods developed poses an intriguing question: if it is possible to obtain the correct answer with a resolution III experiment and a fractional precision of 1 requiring only 44 function evaluations, why would additional function evaluations be necessary? It was suggested in Chapter 3 that a resolution IV experiment buys additional reliability and confidence in the results, and a smaller fractional precision limits the allowable inaccuracy. Yet this example appears to show that the central issue is not accuracy but confidence in the results. Additional function evaluations may find better extrema. A resolution IV exper-

iment combined with a smaller fractional precision, however, is less likely to miss the correct extrema.

In Section 4.2.2, the methods introduced in Chapter 3 for mapping  $\alpha$ -cuts from the DVS to the PVS were demonstrated on quadratic approximations to the bending and torsional stiffnesses calculated by finite element analysis. The two most non-linear design variables *cross-brace area* and *B pillar location* were selected. Despite the non-linearity of  $K_B$  in *B pillar location*, the correct extrema were found when a fractional precision of 0.01 was specified. Moreover, even for a fractional precision of 0.01, *cross-brace area* was sufficiently linear to be approximated by  $\vec{f}'$ . However, the geometry of  $P_\epsilon^d$  was poorly approximated by  $P_\epsilon^{d'}$ . Where  $\vec{f}'$  is sufficiently non-linear that  $P_\epsilon^{d'}$  obtained via linear approximation is inadequate, effective methods to more accurately determine the geometry of  $P_\epsilon^d$  have yet to be developed within the method of imprecision. Developing these methods is an important goal for future research. Nevertheless, this example shows that even where variables are non-linear, the correct extrema can be located and linear approximation can still be valuable in facilitating optimization.

These results demonstrate that a selectively applied linear approximation can be surprisingly effective in evaluating an example finite element model of an automobile body. Finite element models are widely used in industry. The automobile structure design and aircraft engine development examples represent two contrasting design problems from industry. For the automobile structure the design variables were continuous; for engine development cost they were discrete. For the automobile structure example there were two performance variables  $K_B$  and  $K_B$ ; for the engine development example only development cost was quantified. The verification of the algorithms for method of imprecision calculations on realistic design problems is a key contribution of this thesis.

## Chapter 5

### Interpreting Imprecision

Design preferences  $\mu_{d_i}$  and functional requirements  $\mu_{p_j}$  were defined in Chapter 2. Yet the specification of these preferences is significantly more difficult than defining what they represent. Moreover, as will be discussed in this section, the form in which preferences should be specified is dependent on the algorithms used.

#### 5.1 Specifying Preferences

The algorithms described in Chapter 3 for mapping the combined design preference  $\mu_d$  from the DVS to the PVS rely on approximating  $\mu_d(\vec{d})$  as  $\alpha$ -cut sets  $D_{\alpha_1}^d, \dots, D_{\alpha_M}^d$ , defined in Equation (3.5). This discretizes the preference function  $\mu_d(\vec{d})$  into  $M$  preference levels  $\alpha_1, \dots, \alpha_M$ . Mapping  $M$   $\alpha$ -cut sets instead of a continuous preference function leads to a computationally tractable algorithm. But appropriate values for  $\alpha_1, \dots, \alpha_M$  have not yet been defined.

Recall from Section 3.3 that aggregating discretized individual design preferences  $\mu_{d_1}, \dots, \mu_{d_n}$  with an aggregation function other than  $\mathcal{P}_{\min}$  creates additional level sets. These additional level sets occur at intermediate preference levels corresponding to the aggregation of dissimilar  $\alpha$ . Hence if individual design preferences are discretized into  $M_I$  levels of preference  $\alpha_1, \dots, \alpha_{M_I}$ , then the combined design preference will be discretized into  $M_D$  levels, where  $M_D \geq M_I$ . The following discussion applies equally to functional requirements that are discretized into the same  $M_I$  levels of preference  $\alpha_1, \dots, \alpha_{M_I}$ . The combined functional requirement will then

be discretized into  $M_P$  levels where  $M_P \geq M_I$ .  $M_P$  is not necessarily equal to  $M_D$ .

The preference levels  $\alpha_1, \dots, \alpha_{M_I}$  for the individual design preferences are automatically propagated through the aggregation function  $\mathcal{P}$  because of idempotency. The additional preference levels result from  $\mathcal{P}$  defined for the number of design variables  $n$  operating on heterogeneous combinations of the preference values  $\alpha_1, \dots, \alpha_{M_I}$ . For the design preference aggregation example described in Section 3.3 with  $n = 2$  design variables and  $M_I = 3$  levels of preference, three potentially distinct preference levels are created:  $\alpha_{1,2} = \mathcal{P}(\alpha_1, \alpha_2)$ ,  $\alpha_{1,3} = \mathcal{P}(\alpha_1, \alpha_3)$ , and  $\alpha_{2,3} = \mathcal{P}(\alpha_2, \alpha_3)$ . Generalizing to arbitrary  $n$ , there are four situations for which an intermediate preference level  $\alpha_{i,\dots,k}$  where  $\alpha_i < \dots < \alpha_k$  would not be distinguished from  $\alpha_1, \dots, \alpha_{M_I}$ :

1.  $\mathcal{P}$  is such that  $\mathcal{P}(\alpha_i, \dots, \alpha_k) = \alpha_i$ , e.g.,  $\mathcal{P}_{\min}$ ,
2.  $\mathcal{P}$  is such that  $\mathcal{P}(\alpha_i, \dots, \alpha_k) = \alpha_k$ , e.g.,  $\mathcal{P}_{\max}$  except when  $\alpha_i \approx 0$ ,
3.  $\alpha_i$  is zero or sufficiently close to zero that annihilation and continuity require that  $\mathcal{P}(\alpha_i, \dots, \alpha_k) \approx \alpha_i$ ,
4.  $\mathcal{P}(\alpha_i, \dots, \alpha_k) = \alpha_j$  by coincidence,  $\alpha_i < \alpha_j < \alpha_k$ .

Although it might seem that larger  $M_I$  and hence finer discretization would lead to more accurate calculations, there are two compelling reasons why fewer  $\alpha$ -cuts may be better. First, each additional level set required to describe  $\mu_d(\vec{d})$  must be separately mapped onto the PVS, incurring additional function evaluations. Each additional  $\alpha$ -cut defined for the individual design preferences potentially creates as many new level sets for  $\mu_d(\vec{d})$  as there are new combinations of  $\alpha$ . Hence adding  $\alpha$ -cuts, in general, increases computational effort significantly faster than linearly.

Second, the accuracy with which design preferences are represented should not exceed the accuracy with which they can be elicited, or the accuracy with which the results can be interpreted. Design problems with fewer than four design variables and three performance variables are unlikely to pose significant challenges for designers in industry. Yet the single approximated  $\alpha$ -cut in the PVS shown in Figure 3.8, for four design variables and three performance variables, illustrates the difficulty in visualizing multi-dimensional  $\alpha$ -cut sets. Imagine three such regions nested

in the three-dimensional PVS. Then imagine the same three regions modified after aggregation with  $\mu_p$  and mapped back onto the four-dimensional DVS. There is insufficient motivation to calculate additional  $\alpha$ -cut sets that will add information of diminishing value.

How many preference levels  $\alpha_1, \dots, \alpha_{M_I}$  should be defined and what values should be specified? Since the computational algorithm requires individual design preference functions  $\mu_{d_i}$  to be discretized into  $\alpha$ -cut intervals  $[d_{i_{\min}}^{\alpha_1}, d_{i_{\max}}^{\alpha_1}], \dots, [d_{i_{\min}}^{\alpha_{M_I}}, d_{i_{\max}}^{\alpha_{M_I}}]$ , the  $\mu_{d_i}$  should be directly specified, by the designer, in terms of these  $\alpha$ -cut intervals. The preference levels  $\alpha_1, \dots, \alpha_{M_I}$  should therefore be chosen to be meaningful to the designer. The two most important  $\alpha$ -cut intervals are at  $\alpha = 1$ , corresponding to ideal variable values, and  $\alpha = \epsilon$  ( $0 < \epsilon \ll 1$ ), corresponding to barely acceptable variable values. A preference of one indicates an ideal variable value. The  $\alpha$ -cut interval at  $\alpha = 1$  identifies the ideal or target range of values for the variable. These values fully satisfy the considerations that are represented by the preference function. A preference of zero indicates an unacceptable variable value, which can only produce an unacceptable design which fails to satisfy the relevant considerations. An infinitesimal yet non-zero preference  $\epsilon$  indicates a barely acceptable variable value. The  $\alpha$ -cut interval at  $\alpha = \epsilon$  identifies the largest acceptable range of values for the variable. Values outside this range have zero preference and are thus unacceptable.

The barely acceptable  $\alpha$ -cut set defined by the combined design preference is  $D_\epsilon^d$ .  $D_\epsilon^d$  defines the set of designs that are acceptable with respect to design preferences, *i.e.*, with respect design considerations: the unquantified aspects of design performance not represented by performance variables. This specifically excludes preferences arising from performance considerations: the quantified aspects of performance represented by performance variables.  $D_\epsilon^d$  includes all designs that are minimally acceptable relative to design preferences only. Applying functional requirements  $\mu_{p_j}$  will eliminate designs  $\vec{d} \in D_\epsilon^d$  with unacceptable performances and identify a subset of  $D_\epsilon^d$  that describes acceptable designs relative to all specified preferences.

The infinitesimal preference  $\epsilon$  is a special case. It is the smallest value of preference for which a useful  $\alpha$ -cut can be defined. An  $\alpha$ -cut at  $\alpha = 0$  would include the entire DVS. Additionally, annihilation and continuity ensure that aggregating a preference of  $\epsilon$  with any other preference values will result in an infinitesimal preference which is effectively equal to  $\epsilon$ : aggregations involving  $\epsilon$  result in  $\epsilon$ . Intermediate preference levels that aggregate an  $\alpha$ -cut at  $\alpha = \epsilon$  with other  $\alpha$ -cuts, will have preference equal to  $\epsilon$ , *i.e.*, no new preference level is created. Indeed, given that new preference levels are created by aggregating dissimilar  $\alpha$ -cuts, monotonicity and idempotency ensure that there are no new preference levels created between  $\epsilon = \alpha_1$  and the second lowest  $\alpha$ -cut preference  $\alpha_2$ . Preference levels below  $\alpha_2$  can only be generated by aggregation involving a preference less than  $\alpha_2$ . The only  $\alpha$ -cut preference below  $\alpha_2$  is  $\epsilon$ , and aggregations involving  $\epsilon$  result in  $\epsilon$ . Thus defining an  $\alpha$ -cut at  $\alpha = \epsilon$  is computationally efficient.

A basic implementation of the method of imprecision would use only two  $\alpha$ -cuts  $\alpha_1 = \epsilon$  and  $\alpha_2 = 1$ . This requires the designer to specify acceptable intervals for each variable which barely satisfy design and performance considerations, and ideal intervals for each variable which fully satisfy design and performance considerations. These two  $\alpha$ -cuts have been chosen as being the most naturally defined. The concept of a range of acceptable values and a range of ideal values for a variable is not new to engineers in industry. It is the careful specification of these ranges with respect to clearly identified design and performance considerations, on design and performance variables, and the explicit calculation of how they combine on the PVS and the DVS, that is innovative.

As discussed above, aggregation cannot create new preference levels between  $\epsilon$  and 1. Mapping preferences between the DVS and the PVS also does not create new preference levels. Although functional requirements can be specified as functions with continuous preference, these functions are necessarily combined with design preferences that are represented simply as two  $\alpha$ -cuts at  $\alpha_1 = \epsilon$  and  $\alpha_2 = 1$ . In the absence of design preference information at intermediate levels of preference, calculated results are reliable only at preference levels near these two extremes.

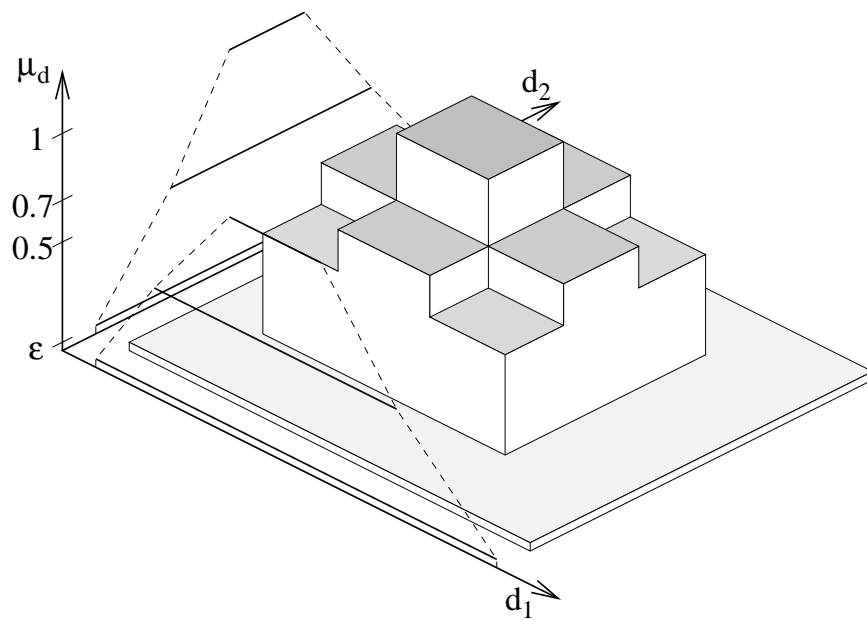
Therefore, it is assumed that functional requirements are also specified only at  $\alpha_1 = \epsilon$  and  $\alpha_2 = 1$ .

Specifying only two  $\alpha$ -cuts  $\alpha_1 = \epsilon$  and  $\alpha_2 = 1$  eliminates the difference between aggregation functions. Aggregations involving the same preference  $\alpha_k$  return  $\alpha_k$  by idempotency. All other aggregations involve  $\epsilon$  and result in  $\epsilon$ : these will be subsets of the  $\alpha$ -cut at  $\alpha_1 = \epsilon$  and need not be calculated. Relatively few function evaluations are required. All design-appropriate aggregation functions reduce to the intersection of intervals, regardless of weights. Thus approximating preferences as only two  $\alpha$ -cuts obviates the identification of aggregation functions or even an aggregation hierarchy. Barely acceptable preference intervals  $\alpha_1 = \epsilon$  and ideal intervals  $\alpha_2 = 1$  are simply intersected (separately for each  $\alpha$ ) to aggregate. Moreover, it is easier to specify a set of ideal and acceptable intervals than a continuous functional requirement, and the results are more easily interpreted. Thus the propagation of ideal and acceptable intervals can be used as a preliminary analysis of the design problem in order to sketch the boundaries of the design space while also searching for an ideal set of designs.

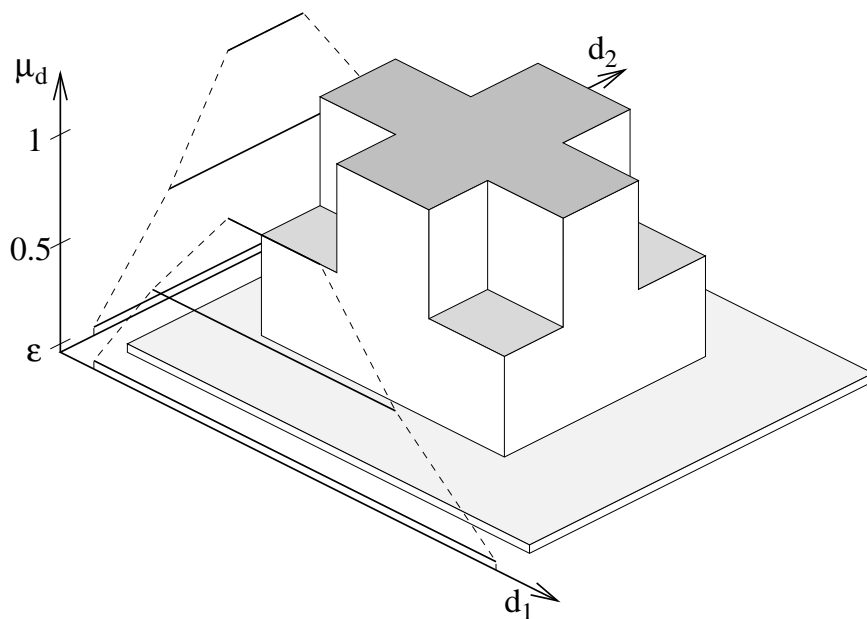
A more complete implementation would add an intermediate  $\alpha$ -cut. A natural choice would be  $\alpha = 0.5$ . A preference of 0.5 lies halfway between unacceptable and ideal. Membership of 0.5 in a fuzzy set implies membership of 0.5 in its complement. Analogously, a preference of 0.5 indicates a variable value that is equally well (or equally poorly) described as ideal or as unacceptable. The designer or customer is neither satisfied nor dissatisfied. The  $\alpha$ -cut interval at  $\alpha = 0.5$  identifies a neutral or indifferent range of values for the variable. This implies that a preference above 0.5 expresses positive satisfaction, such that the customer or designer considers the variable value to be desirable or better than the norm. A preference below 0.5 expresses negative satisfaction, *i.e.*, dissatisfaction, such that the customer or designer considers the variable value to be undesirable or worse than the norm.

Thus the three  $\alpha$ -cuts can be characterized:

- $\alpha_1 = \epsilon$  indicates undesirable yet barely acceptable variable values,
- $\alpha_2 = 0.5$  indicates neutral or indifferent variable values,



**Figure 5.1** Aggregating three  $\alpha$ -cuts: level sets for  $\mu_d = \mathcal{P}_\Pi(\mu_{d_1}, \mu_{d_2})$ .



**Figure 5.2** Aggregating three  $\alpha$ -cuts: level sets for  $\mu_d = \mathcal{P}_{\max'}(\mu_{d_1}, \mu_{d_2})$ .

- $\alpha_3 = 1$  indicates (desirable) ideal or target variable values.

As before, no new preference levels are created between  $\alpha_1$  and  $\alpha_2$  and any aggregation involving  $\alpha_1 = \epsilon$  results in  $\epsilon$ . Aggregations involving  $\alpha_2 = 0.5$  and  $\alpha_3 = 1$ , however, can create new intermediate preference levels depending on the aggregation function  $\mathcal{P}$ . Therefore, increasing the number of  $\alpha$ -cuts from two to three potentially increases the number of design preference level sets and hence the computational effort, by more than 50%. The number of new preference levels created depends on the form of the aggregation function  $\mathcal{P}$ : the non-compensating trade-off  $\mathcal{P}_{\min}$  and the design-appropriate maximal trade-off  $\mathcal{P}_{\max'}$  do not create intermediate preference levels; other aggregation functions do. Figure 5.2 shows the level sets that result from using  $\mathcal{P}_{\max'}$  to aggregate individual design preference  $\alpha$ -cuts at three levels:  $\alpha_1 = \epsilon$ ,  $\alpha_2 = 0.5$ , and  $\alpha_3 = 1$ . There are  $n = 2$  design variables. Although the combined design preference level set at  $\alpha_3 = 1$  is not rectangular, no new preference levels are created. Figure 5.1 shows the level sets that result from using  $\mathcal{P}_{\Pi}$  to aggregate the same individual design preferences. A new preference level at  $\mu_d = \mathcal{P}_{\Pi}(0.5, 1) \approx 0.7$  is created. This intermediate level set describes the combined preference where the individual design preferences are 0.5 for one design variable and 1 for the other. The effect of using a different aggregation function  $\mathcal{P}$  is to vary the height of the intermediate level set between the limits defined by  $\mathcal{P}_{\min}$  (Figure 3.5) and  $\mathcal{P}_{\max'}$  (Figure 5.2).

The number of potentially new intermediate preference levels is equal to the number of unique (non-ordinal) combinations of  $n$  preferences chosen from  $\{\alpha_2, \alpha_3\}$ , excluding  $(\alpha_2, \dots, \alpha_2)$  and  $(\alpha_3, \dots, \alpha_3)$ . There are  $n - 1$  such combinations. Including the new  $\alpha$ -cut at  $\alpha_3 = 1$  brings the total number of new preference levels to  $n$ : the increase in  $M_D$  and complexity grows linearly with the number of design variables  $n$ . For the example given above with two design variables, the third  $\alpha$ -cut increases the maximum number of preference levels from two to four (Figure 5.1). If there are eight design variables, however, the increase is from two to ten: a fivefold jump in complexity. Adding further  $\alpha$ -cuts exacts an even stiffer penalty. The number of unique combinations of  $n$  preferences chosen from  $\{\alpha_2, \alpha_3, \alpha_4\}$ , excluding  $(\alpha_2, \dots, \alpha_2)$

and  $(\alpha_3, \dots, \alpha_3)$  is  $\frac{1}{2}(n+1)(n+2) - 2$ . Hence the total number of potentially new preference levels added by a fourth  $\alpha$ -cut is  $\frac{1}{2}n(n+1)$ . For eight design variables, a fourth  $\alpha$ -cut potentially adds 36 level sets to bring the total to  $M_D = 46$ .

Yet the proliferation of new preference levels brings the ability to reflect the effects of different aggregation functions. The intermediate preference levels encode different trade-offs between design variables that are not modeled in the simple two  $\alpha$ -cut implementation. The third  $\alpha$ -cut is necessary to fully implement the method of imprecision.

## 5.2 A Scenario for Implementation

This section develops the electric vehicle design problem from Section 2.7 in order to present a possible scenario for implementing the method of imprecision in an industrial setting. The scenario is that the company produces a variety of electric vehicles based on a common basic chassis. A new vehicle intended for short distances (less than 50 miles) is to be developed. The vehicle will be used within the neighborhood and for commuting moderate distances. It is assumed that recharging will be available at home and at work. The vehicle will be a four-door compact able to seat four adults or two adults and three children. Target buyers are under 40, environmentally conscious, suburban or city residents, male or female, with household incomes between \$30,000 and \$100,000. To begin with, a preliminary analysis of the design problem is required. Such a preliminary analysis can be provided by a minimal implementation of the method of imprecision which uses design preference and functional requirement  $\alpha$ -cuts at only two levels,  $\epsilon$  and 1, as discussed in Section 5.1.

The first step is to identify design and performance variables. Design variables are the attributes under active consideration that distinguish alternatives that are to be evaluated separately. Performance variables are those aspects of design performance that are to be quantified. The means of evaluating  $p_j = f_j(\vec{d})$  should also be defined. Given that this is a new variant in a family of similar electric vehicles,

these evaluation tools should be readily available. The performance considerations that will be represented by functional requirements on the performance variables and the design considerations that will be represented by design preferences on the design variables must be identified. The design considerations for this problem were discussed in Section 2.7. Performance considerations are typically obviously related to the performance variables. *Range* and *cost* require little explanation. The customer's preferences on these variables are direct and do not need to be interpreted. Because it is widely reported and frequently the only numerical measure of vehicle performance that the customer is aware of, *0-60 time* has particular significance. *Weight* is a measure of the efficiency of the design that impacts vehicle handling. *Bending stiffness*  $K_B$  and *torsional stiffness*  $K_T$  are indicators of NVH performance [12], as well as vehicle handling.

The next step is to specify individual design preference and functional requirement intervals at  $\alpha_1 = \epsilon$  and  $\alpha_2 = 1$ . These correspond to the two separate ranges of barely acceptable and ideal values:

- for the design variables based on appropriate design considerations, and
- for the performance variables based on appropriate performance considerations.

For such a minimal implementation, the  $n$ -cubic combined design preference  $\alpha$ -cuts  $D_\epsilon^d$  and  $D_1^d$  fully describe  $\mu_d(\vec{d})$ . No non- $n$ -cubic  $\alpha$ -cuts at intermediate preferences are created. Experiment design is used to calculate regression coefficients for  $\vec{f}$  on  $D_\epsilon^d$  and construct a linear approximation  $\vec{f}^{\vec{f}}$  in acceptably linear variables. The evaluated points and  $\vec{f}^{\vec{f}}$  are selectively applied to facilitate optimization and to calculate the full geometry of the approximate combined design preference  $\alpha$ -cuts on the PVS  $P_\epsilon^{d'}$  and  $P_1^{d'}$ .  $P_\epsilon^{d'}$  and  $P_1^{d'}$  are then intersected with the combined functional requirement  $\alpha$ -cuts  $P_\epsilon^p$  and  $P_1^p$  to obtain the  $\alpha$ -cut representations of  $\mu_o(\vec{p})$ . Aggregation is reduced to the intersection of intervals separately at  $\alpha = \epsilon$  and  $\alpha = 1$ . The results of this first calculation are two approximate  $\alpha$ -cut sets at  $\epsilon$  and 1 that indicate the acceptable and ideal performances based on the complete

set of specified preferences. It is possible, even probable, that the  $\alpha$ -cut set at a preference of 1 is the empty set, *i.e.*, there are no performances that simultaneously achieve ideal values for all design and performance variables. Examination of  $P_1^{d'}$  and  $P_1^p$  may indicate which functional requirements are not fully satisfied. If the  $\alpha$ -cut set at a preference of  $\epsilon$  is also the empty set, then some adjustment of the originally specified preferences is necessary since no acceptable performances exist.

The discussion has, so far, referred to a single designer. Design problems are typically solved by design teams. Different team members will have different areas of expertise as well as different opinions and preferences. Furthermore, group decision-making in design tends to involve bargaining or negotiation [55]. Negotiation is not easily formalized in a design methodology. The relative power, authority, and expertise of the parties involved is amenable neither to formal modeling nor to explicit expression. It is not suggested that the method of imprecision can perform the trade-offs between the differing preferences of team members on the same variable. It is even problematic to suggest that the method be used as a common framework to resolve the different preferences expressed by team members on their particular segments of the design. Bargaining would be manifested in the lowballing or highballing of  $\alpha$ -cut intervals. The form of the aggregation hierarchy and relative importance weightings would also be subject to negotiation.

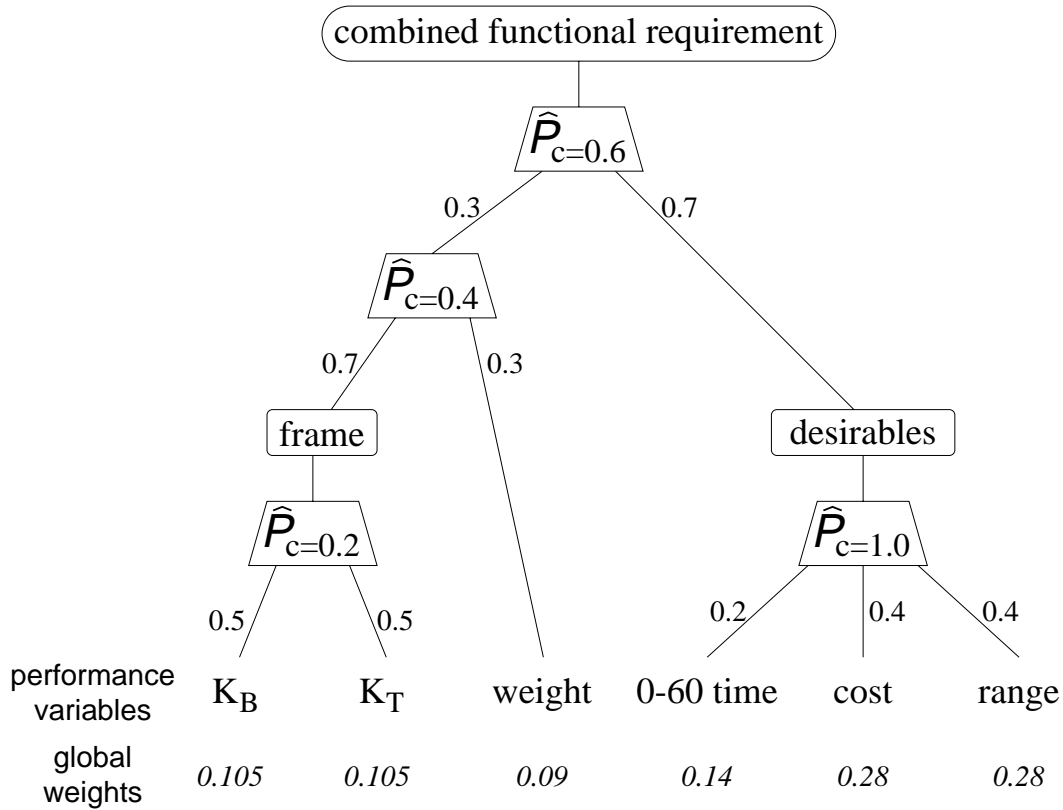
One alternative would be to appoint a single facilitator, who would need to have enough authority and experience to weigh the different opinions of the team members. Constructing the method of imprecision model would be the responsibility of the facilitator, but the preferences specified and the aggregation hierarchies identified would be those of the design team. In this way, the resolution of conflicting opinions remains an informal process, even though it falls principally on the shoulders of one person: the facilitator. Yet this may discourage extreme bargaining behavior: extreme opinions risk being dismissed by the facilitator. Borrowing from the idea of pendulum arbitration, this could be formalized by requiring the facilitator to choose only one side at a time to revise its opinions towards producing a set of designs that are acceptable for all team members. Lowballing or highballing

increases the chance of being chosen. There is thus an incentive to submit information that is realistic instead of exaggerated, as well as a mechanism for building consensus. For large groups, several participants could be required to revise their opinions during each iteration, instead of just one.

For the implementation scenario presented, this form of arbitrated iteration would be used to determine a set of collectively specified preferences that intersect to give a non-empty set of acceptable performances. It may even be possible to further modify preferences until a non-empty set of ideal performances is found. These preliminary calculations are best performed at minimal levels of accuracy, since it is only necessary to establish whether a non-empty set of performances exist at preferences of  $\epsilon$  and 1. Once such a set has been shown to exist, it can be more precisely determined.

The next step is to map the combined functional requirement onto the DVS. This relies on the linear approximation  $\vec{f}$  constructed for the forward mapping of design preferences onto the PVS. The approximate combined functional requirement  $\alpha$ -cuts on the DVS  $D_\epsilon^{p'}$  and  $D_1^{p'}$  obtained are then intersected with the combined design preference  $\alpha$ -cuts  $D_\epsilon^d$  and  $D_1^d$ . The final results of this backwards calculation are two approximate  $\alpha$ -cut sets at  $\epsilon$  and 1 that indicate the barely acceptable and ideal designs based on the complete set of specified preferences. These overall preference  $\alpha$ -cut sets in the DVS correspond to the overall preference  $\alpha$ -cut sets in the PVS. The shape of these sets indicate the full range of acceptable designs, a restricted range of ideal designs, and the performances achievable by each set of designs. Particular designs  $\vec{d}$  on the boundaries of the  $\alpha$ -cut sets in the DVS will have been evaluated and hence can be matched to their corresponding performances  $\vec{p} = \vec{f}(\vec{d})$ .

The purpose of this preliminary analysis is to construct a basic model of the design problem with design and performance variables explicitly related to design and performance considerations, and to settle on a set of acceptable and ideal intervals that intersect to produce at least a non-empty set of acceptable designs and performances, and possibly also a non-empty set of ideal designs and performances. The  $\alpha$ -cut overall preference sets at  $\epsilon$  demarcate the limits of the possible designs and



**Figure 5.3** Functional requirement aggregation hierarchy for an electric vehicle design.

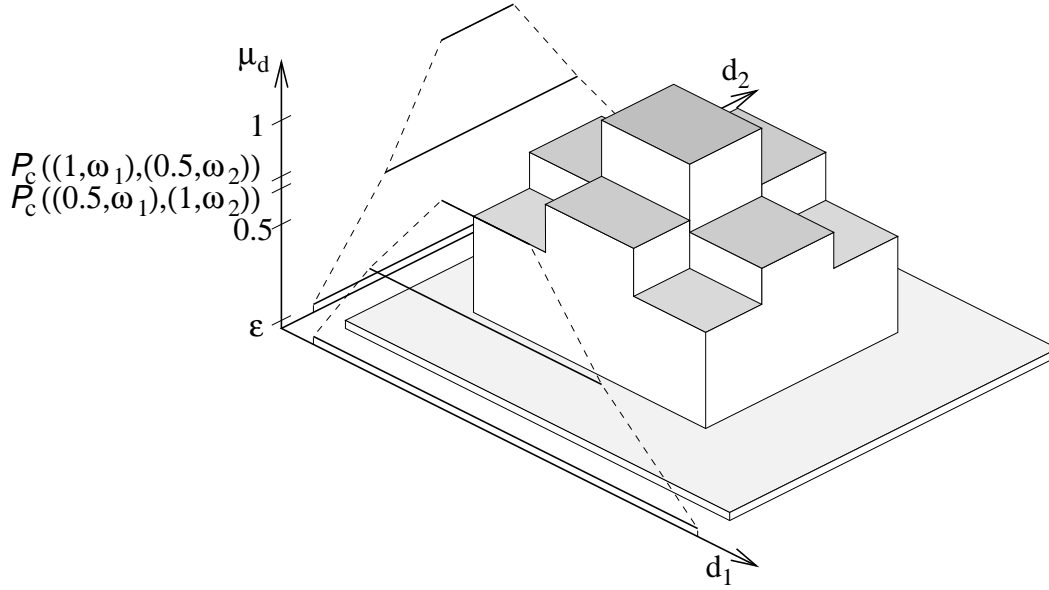
performances as in Toyota's design and development process [65]. These boundaries can be communicated to downstream engineering groups in order to facilitate concurrency in the design engineering process. Moreover, if non-empty sets of ideal designs and performances exist, these can provide additional information. The method of imprecision can provide not only set-based information, but set-based information that distinguishes more than one level of preference.

The next stage of the design will require a more precise analysis. The purpose is now to refine the set of possible designs by evolving a more precise set of preferences that distinguish promising designs. Moving from the minimal two  $\alpha$ -cut implementation of the method of imprecision to the more complete three  $\alpha$ -cut implementation (Section 5.1) requires the determination of design preference and

functional requirement aggregation hierarchies. The design preference hierarchy is shown in Figure 2.3 and has already been discussed. The functional requirement hierarchy is shown in Figure 5.3. *Bending stiffness*  $K_B$  and *torsional stiffness*  $K_T$  are both frame parameters and are naturally grouped together. They do not compensate strongly for each other because they measure two different modes of the vehicle's dynamics that together determine vehicle handling and perceived comfort and quality. *Range*, *cost*, and *0–60 time* are performance variables of direct interest to the customer that can be traded-off with a high degree of compensation. They are labeled as “desirables.” All the possible trade-offs among these three variables that increase one or more preferences but reduce the others yield marketable vehicles: a slow, short range, but cheap vehicle; an expensive, slow, but long range vehicle; an expensive, short range, but fast vehicle; an even more expensive, but fast and long range vehicle, *etc.* A fully compensating trade-off ( $c = 1$ ) is specified. *Weight* is less easily located in the hierarchy. It is neither directly of interest to the customer, nor is it purely a frame parameter. A difficulty is that the most obvious consequences of *weight* are *range* and *0–60 time*, which are already performance variables. The performance considerations that remain for *weight* are design efficiency and vehicle handling. Since  $K_B$  and  $K_T$  are better measures of vehicle handling, *weight* becomes a surprisingly unimportant performance variable, with a global importance weight of 0.09. In comparison, *cost* and *range* have global weights of 0.28 each, *0–60 time* has a global weight of 0.14, and  $K_B$  and  $K_T$  each have global weights of 0.105.

The degree of compensation  $c$  is difficult to specify, partly because the parameter  $k$  in Equation (2.10) that defines  $\mathcal{P}_c$  has not been determined. Without a value for  $k$ , only three values of  $c$  are pinned down:  $c = 0$  is non-compensating,  $c = 1$  is fully compensating, and  $c = 2$  is maximally (super-) compensating. Yet even without a complete definition of  $\mathcal{P}_c$ , a more systematic method of specifying  $c$  can be developed.

Figure 5.4 shows individual  $\alpha$ -cut intervals in  $d_1$  and  $d_2$  being aggregated by  $\mathcal{P}_c$ . The design considerations represented by  $\mu_{d_1}$  are more important than those represented by  $\mu_{d_2}$  and hence  $\omega_1 > \omega_2$ . It is apparent that  $\mathcal{P}_c((1, \omega_1), (0.5, \omega_2)) >$



**Figure 5.4** Aggregating three  $\alpha$ -cuts with weights,  $\omega_1 > \omega_2$ .

$\mathcal{P}_c((0.5, \omega_1), (1, \omega_2))$ : ideal values of  $d_1$  matched with neutral values of  $d_2$  are preferred to neutral values of  $d_1$  matched with ideal values of  $d_2$ . This is consistent with  $\mu_{d_1}$  being more important than  $\mu_{d_2}$ . The preference levels  $\mathcal{P}_c((1, \omega_1), (0.5, \omega_2))$  and  $\mathcal{P}_c((0.5, \omega_1), (1, \omega_2))$  are determined by  $\mathcal{P}_c$ , or more specifically the degree of compensation  $c$ . Thus choosing the degree of compensation is, for this three  $\alpha$ -cut implementation, equivalent to choosing the preference levels  $\mathcal{P}_c((1, \omega_1), (0.5, \omega_2))$  and  $\mathcal{P}_c((0.5, \omega_1), (1, \omega_2))$ . This can be framed in terms of two questions:

- If  $d_1$  is ideal and  $d_2$  is neutral, what is your combined preference?
- If  $d_1$  is neutral and  $d_2$  is ideal, what is your combined preference?

These two questions are not independent because only one degree of compensation can be specified. Furthermore, it is necessary to keep in mind the pivotal values of preference given by  $\mathcal{P}_{c=0} = \mathcal{P}_{\min}$ ,  $\mathcal{P}_{c=1} = \mathcal{P}_{\Pi}$ , and  $\mathcal{P}_{c=2} = \mathcal{P}_{\max}$ .  $\mathcal{P}_{c=0}(0.5, 1) = 0.5$  and  $\mathcal{P}_{c=2}(0.5, 1) = 1$ : these functions are independent of weights. Thus the extremes of compensation are clearly given. But as is discussed in Section 2.5, su-

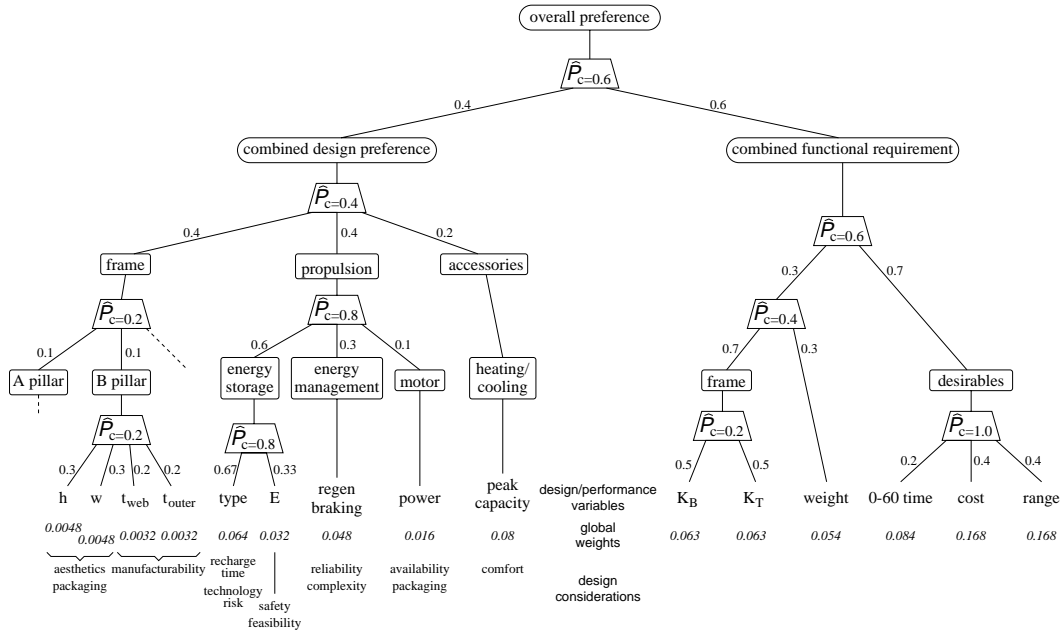


Figure 5.5 Overall preference aggregation hierarchy.

percompensating values of  $c > 1$  imply a willingness to trade-off small gains in one preference for large losses in another. They are thus not suitable for typical design problems, despite satisfying the axioms for design-appropriateness. For most problems, the limits of  $c$  are zero ( $\mathcal{P}_{\min}$ ) and one ( $\mathcal{P}_{\Pi}$ ).  $\mathcal{P}_{c=1}((0.5, \omega_1), (1, \omega_2))$  and  $\mathcal{P}_{c=1}((1, \omega_1), (0.5, \omega_2))$  depend on the ratio of  $\omega_1$  to  $\omega_2$  and are not equal unless  $\omega_1 = \omega_2$ . It is also not intuitively obvious what their values are. For  $\omega_1 = \omega_2$ ,  $\mathcal{P}_{c=1}((0.5, \omega_1), (1, \omega_1)) = 0.71$ . Thus if the degree of compensation is to be determined using this procedure, it is important to evaluate  $\mathcal{P}_{c=1}((0.5, \omega_1), (1, \omega_2))$  and  $\mathcal{P}_{c=1}((1, \omega_1), (0.5, \omega_2))$  in order to pin down the extreme of full compensation. The preference levels  $\mathcal{P}_c((1, \omega_1), (0.5, \omega_2))$  and  $\mathcal{P}_c((0.5, \omega_1), (1, \omega_2))$  that determine  $c$  can then be chosen relative to  $\mathcal{P}_{c=0}$  and  $\mathcal{P}_{c=1}$ . This is still an informal and approximate procedure, but it is an improvement over direct estimation.

After the design preference and functional requirement aggregation hierarchies have been defined, one final aggregation operation needs to be specified:  $\hat{\mu}_o =$

$\widehat{\mathcal{P}}_c((\mu_d, \omega_d), (\mu_p, \omega_p))$ . The combined design preference and the combined functional requirement must themselves be aggregated into the overall preference. Since functional requirements represent performance considerations which are important enough to be quantified as performance variables,  $\mu_p$  is expected to be more important than  $\mu_d$ . The complete preference aggregation hierarchy is shown in Figure 5.5. The relative weights of 0.4 for the combined design preference and 0.6 for the combined functional requirement were chosen to appropriately scale the global weights associated with particular design and performance variables.

The aggregation of individual preferences into the overall preference is now fully defined. Through the arbitration procedure described above, or otherwise, individual design preference and functional requirement intervals at  $\alpha_1 = \epsilon$ ,  $\alpha_2 = 0.5$ , and  $\alpha_3 = 1$  are specified. The discretized design preferences are aggregated via the design preference aggregation hierarchy. The combined design preference  $\alpha$ -cuts obtained are then mapped onto the PVS using the methods described in Chapter 3, some of which were demonstrated in Chapter 4. The individual functional requirements are aggregated via the functional requirement aggregation hierarchy into the combined functional requirement  $\mu_p(\vec{p})$ .  $\mu_d(\vec{p})$  and  $\mu_p(\vec{p})$  are then combined via the final aggregation operation shown in overall preference aggregation hierarchy. The results of this forward calculation are  $\alpha$ -cut approximations to  $\mu_o(\vec{p})$ , the overall preference on the PVS. These  $\alpha$ -cut sets indicate the multi-dimensional performances (*range, cost, 0-60 time, weight,  $K_B$ , and  $K_T$* ) that are achievable by designs at the three principal levels of overall preference  $\epsilon$  (acceptable), 0.5 (neutral), and 1 (ideal), as well as intermediate levels of preference between 0.5 and 1 that indicate desirable designs that trade-off neutral and ideal preferences.

The backward calculation requires reversing the mapping  $\vec{f} : \text{DVS} \rightarrow \text{PVS}$  using the linear approximation  $\vec{f}'$ . The combined functional requirement is mapped onto approximate  $\alpha$ -cuts in the DVS and these  $\alpha$ -cuts are aggregated with the combined design preference via the same overall preference aggregation operation as for the forward calculation. The  $\alpha$ -cut sets obtained approximate  $\mu_o(\vec{d})$ , the overall preference on the DVS. These  $\alpha$ -cuts indicate the sets of designs that correspond to

the principal overall preference levels  $\epsilon$  (acceptable), 0.5 (neutral), and 1 (ideal), as well as intermediate overall preference levels between 0.5 and 1 that indicate designs that trade-off neutral and ideal preferences.

### 5.3 Conclusions

The specification of design preferences  $\mu_{d_i}$  and functional requirements  $\mu_{p_j}$  is significantly more difficult than defining what they represent. Moreover, the form of these specified preferences is dependent on the algorithms used. The algorithms described in Chapter 3 for mapping the combined design preference  $\mu_d$  from the DVS to the PVS rely on approximating  $\mu_d(\vec{d})$  as  $\alpha$ -cut sets  $D_{\alpha_1}^d, \dots, D_{\alpha_{M_I}}^d$ . This discretizes the preference function  $\mu_d(\vec{d})$  into  $M_I$  preference levels  $\alpha_1, \dots, \alpha_{M_I}$ .

Aggregating discretized preference functions with an aggregation function other than  $\mathcal{P}_{\min}$  creates additional non- $n$ -cubic level sets (Section 3.3). Although it might seem that finer discretization would lead to more accurate calculations, there are two compelling reasons why fewer  $\alpha$ -cuts may be better.

1. Each additional level set required to describe  $\mu_d(\vec{d})$  must be separately mapped onto the PVS, incurring additional function evaluations. Each additional  $\alpha$ -cut defined for the individual design preferences potentially creates as many new level sets for  $\mu_d(\vec{d})$  as there are new combinations of  $\alpha$ .
2. The accuracy with which design preferences are represented should not exceed the accuracy with which they can be elicited, or the accuracy with which the results can be interpreted. The difficulty of visualizing even one  $\alpha$ -cut set in a four-dimensional DVS suggests that calculating additional  $\alpha$ -cut sets will only add information of diminishing value.

The two most important  $\alpha$ -cut intervals are at  $\alpha = 1$ , corresponding to ideal variable values, and  $\alpha = \epsilon$  ( $0 < \epsilon \ll 1$ ), corresponding to barely acceptable variable values. The  $\alpha$ -cut interval at  $\alpha = 1$  identifies the ideal or target range of values for the variable. A preference of zero indicates an unacceptable variable value, which

can only produce an unacceptable design which fails to satisfy the relevant considerations. An infinitesimal yet non-zero preference  $\epsilon$  indicates a barely acceptable variable value. The  $\alpha$ -cut interval at  $\alpha = \epsilon$  identifies the largest acceptable range of values for the variable. Anihilation and continuity ensure that aggregating a preference of  $\epsilon$  with any other preference values results in an infinitesimal preference which is effectively equal to  $\epsilon$ . Monotonicity and idempotency ensure that aggregation cannot create new preference levels between  $\alpha_1 = \epsilon$  and the second lowest  $\alpha$ -cut preference  $\alpha_2$ . Thus defining an  $\alpha$ -cut at  $\alpha = \epsilon$  is computationally efficient.

A basic implementation of the method of imprecision suitable for preliminary analyses would use only two  $\alpha$ -cuts  $\alpha_1 = \epsilon$  and  $\alpha_2 = 1$ . No new preference levels can be created between  $\epsilon$  and 1 and the difference between aggregation functions is eliminated: aggregation is reduced to the intersection of intervals. It is unnecessary to identify aggregation functions or even an aggregation hierarchy. Both design preferences and functional requirements are represented as acceptable and ideal intervals. Such an implementation requires relatively few function evaluations. Moreover, it is easier to specify a set of ideal and acceptable intervals than a continuous functional requirement, and the results are more easily interpreted.

A more complete implementation would use three  $\alpha$ -cuts:

- $\alpha_1 = \epsilon$  indicates undesirable yet barely acceptable variable values.
- $\alpha_2 = 0.5$  indicates neutral or indifferent variable values.
- $\alpha_3 = 1$  indicates (desirable) ideal or target variable values.

As before, no new preference levels are created between  $\alpha_1$  and  $\alpha_2$  and any aggregation involving  $\alpha_1 = \epsilon$  results in  $\epsilon$ . The number of potentially new intermediate preference levels, *i.e.*, the number of unique (non-ordinal) combinations of  $n$  preferences chosen from  $\{\alpha_2, \alpha_3\}$  excluding  $(\alpha_2, \dots, \alpha_2)$  and  $(\alpha_3, \dots, \alpha_3)$ , is  $n - 1$ . Including the new  $\alpha$ -cut at  $\alpha_3 = 1$  brings the total number of new preference levels to  $n$ : the *increase* in complexity grows linearly with the number of design variables  $n$ . For eight design variables the maximum number of preference levels increases from two to ten: a fivefold jump in complexity. Adding further  $\alpha$ -cuts exacts an even stiffer

penalty. The total number of potentially new preference levels added by a fourth  $\alpha$ -cut is  $\frac{1}{2}n(n+1)$ .

Yet the new preference levels that accompany the third  $\alpha$ -cut also bring the ability to model different aggregation functions. The intermediate preference levels encode different trade-offs between design variables that are ignored in the basic two  $\alpha$ -cut implementation. The third  $\alpha$ -cut is necessary to fully implement the method of imprecision.

In Section 5.2 the electric vehicle design problem from Section 2.7 was used to present a possible scenario for implementing the method of imprecision. The scenario involved the development of a new variant in a family of electric vehicles based on a common basic chassis. The steps described in implementing the method of imprecision as discussed in this thesis are as follows:

1. Conduct a preliminary analysis of the design problem using a minimal implementation of the method of imprecision ( $M_I = 2$ ).
  - (a) Identify design and performance variables and define the mappings  $f_j(\vec{d})$ . Identify design and performance considerations corresponding to each variable.
  - (b) Specify individual design preference and functional requirement intervals at  $\alpha_1 = \epsilon$  (acceptable) and  $\alpha_2 = 1$  (ideal).
  - (c) Aggregate individual design preference intervals into  $D_\epsilon^d$  and  $D_1^d$  using the Cartesian product.
  - (d) Use experiment design to calculate regression coefficients for  $\vec{f}$  on  $D_\epsilon^d$  and construct a linear approximation  $\vec{f}^{\vec{d}}$  in acceptably linear variables. Selectively apply the evaluated points and  $\vec{f}^{\vec{d}}$  to facilitate optimization and to calculate the full geometries of the approximate combined design preference  $\alpha$ -cuts on the PVS  $P_\epsilon^{d'}$  and  $P_1^{d'}$ .
  - (e) Aggregate individual functional intervals into  $P_\epsilon^p$  and  $P_1^p$  using the Cartesian product.

- (f) Intersect  $P_\epsilon^{d'}$  and  $P_1^{d'}$  with the combined functional requirement  $\alpha$ -cuts  $P_\epsilon^p$  and  $P_1^p$  to obtain the  $\alpha$ -cut representations of  $\mu_o(\vec{p})$  at  $\epsilon$  and 1: the acceptable and ideal performances.
  - (g) If no acceptable performances exist or if acceptable or ideal performances are otherwise not satisfactory, return to step 1(b) and revise the preferences specified.
  - (h) Use  $\vec{f}'$  to calculate the  $\alpha$ -cuts  $D_\epsilon^{p'}$  and  $D_1^{p'}$  on the DVS, which approximate the pre-images of  $P_\epsilon^p$  and  $P_1^p$ , the combined functional requirement  $\alpha$ -cuts on the PVS.
  - (i) Intersect  $D_\epsilon^{p'}$  and  $D_1^{p'}$  with  $D_\epsilon^d$  and  $D_1^d$  to obtain the  $\alpha$ -cut representations of  $\mu_o(\vec{d})$  at  $\epsilon$  and 1: the acceptable and ideal designs.
  - (j) If the calculated sets of acceptable or ideal designs require adjustment, return to step 1(b) and revise the preferences specified.
2. Perform a more precise analysis using a more complete  $M_I = 3$   $\alpha$ -cut implementation in order to refine the set of possible designs and identify promising designs.
- (a) Determine the design preference and functional requirement aggregation hierarchies, and in particular importance weightings and degree of compensation for aggregation operations.
  - (b) Determine the overall preference aggregation operation  $\widehat{\mathcal{P}}_c((\mu_d, \omega_d), (\mu_p, \omega_p))$ .
  - (c) Specify individual design preference and functional requirement intervals at  $\alpha_1 = \epsilon$  (acceptable),  $\alpha_2 = 0.5$  (neutral), and  $\alpha_3 = 1$  (ideal).
  - (d) Aggregate individual design preference intervals into  $M_D$   $\alpha$ -cuts  $D_\epsilon^d, D_{0.5}^d, \dots, D_1^d$  via the design preference aggregation hierarchy.
  - (e) Use experiment design to (re-)calculate regression coefficients for  $\vec{f}'$  on  $D_\epsilon^d$  and construct a linear approximation  $\vec{f}''$  in acceptably linear variables. Selectively apply the evaluated points and  $\vec{f}''$  to facilitate optimization and to calculate the full geometries of the approximate combined design preference  $\alpha$ -cuts on the PVS  $P_\epsilon^{d'}, P_{0.5}^{d'}, \dots, P_1^{d'}$ .

- (f) Aggregate individual functional requirement intervals into  $M_P$   $\alpha$ -cuts  $P_\epsilon^p, P_{0.5}^p, \dots, P_1^p$  via the functional requirement aggregation hierarchy.
- (g) Aggregate  $P_\epsilon^{d'}, P_{0.5}^{d'}, \dots, P_1^{d'}$  with the combined functional requirement  $\alpha$ -cuts  $P_\epsilon^p, P_{0.5}^p, \dots, P_1^p$  to obtain an  $\alpha$ -cut representation of  $\mu_o(\vec{p})$ . These  $\mu_o(\vec{p})$   $\alpha$ -cuts indicate sets of performances at different levels of preference: barely acceptable, neutral, ideal, and various levels between neutral and ideal.
- (h) If no acceptable performances exist or if calculated  $\alpha$ -cuts performance sets are otherwise not satisfactory, return to step 2(c) or step 2(a).
- (i) Use  $\vec{f}^i$  to calculate the  $\alpha$ -cuts  $D_\epsilon^{p'}, D_{0.5}^{p'}, \dots, D_1^{p'}$  on the DVS, which approximate the pre-images of  $P_\epsilon^p, P_{0.5}^p, \dots, P_1^p$ , the combined functional requirement  $\alpha$ -cuts on the PVS.
- (j) Aggregate  $D_\epsilon^{p'}, D_{0.5}^{p'}, \dots, D_1^{p'}$  with  $D_\epsilon^d, D_{0.5}^d, \dots, D_1^d$  to obtain an  $\alpha$ -cut representation of  $\mu_o(\vec{d})$ . These  $\mu_o(\vec{d})$   $\alpha$ -cuts indicate sets of designs at different levels of preference: barely acceptable, neutral, ideal, and various levels between neutral and ideal.
- (k) If the calculated  $\alpha$ -cut sets of designs require further refinement, return to step 2(c) or step 2(a).

The difficulty of determining the degree of compensation  $c$  for aggregation operations in steps 2(a) and 2(b) was discussed in the context of the three  $\alpha$ -cut implementation introduced in Section 5.1. An informal procedure for exploring the choice of  $c$  relative to the aggregation functions  $\mathcal{P}_{c=0} = \mathcal{P}_{\min}$ ,  $\mathcal{P}_{c=1} = \mathcal{P}_{\Pi}$ , and  $\mathcal{P}_{c=2} = \mathcal{P}_{\max}$  was presented.

For group decision-making, specifying preferences (1(b) and 2(c)) and determining the aggregation hierarchies (2(a) and 2(b)) subject to bargaining or negotiation. One suggested solution is to appoint a facilitator, who would weigh the opinions of the members of the design team in determining preferences and aggregation hierarchies. Whenever preferences or hierarchies need to be modified, the facilitator should be required to choose only one participant's opinions for revision. For large

groups, several participants could be chosen for each iteration. This provides an incentive to submit information that is realistic instead of exaggerated, as well as a mechanism for building consensus.

## Chapter 6

### Conclusions

If a man will begin with certainties, he will end with doubts; but if he will be content with doubts, he shall end in certainties.

*Sir Francis Bacon* (1561–1626), “Advancement of Learning” Bk. 1

What I tell you three times is true.

*Lewis Carroll* (1832–1898), “The Hunting of the Snark”

The method of imprecision represents design imprecision through the customer’s preferences on relevant aspects of design performance.

- Functional requirements model the customer’s direct preference on performance variables based on *performance considerations*: the quantified aspects of design performance represented by performance variables.
- Design preferences model the customer’s anticipated preference on design variables based on *design considerations*: the unquantified aspects of design performance not represented by performance variables.

The precise differentiation between design and performance variables, between design preferences and functional requirements, and between design and performance considerations, is the first key contribution of this thesis. The innovation in distinguishing design considerations is that it provides a clearly defined formal structure

for representing “soft” issues such as aesthetics and manufacturability and quantifying their consequences.

In Section 2.5, a class of aggregation functions were defined using the weighted root-mean-power family of functions. This family of aggregation functions, which represent the model decision-maker in the method of imprecision, allow a broad range of degrees of compensation and satisfy postulated axioms for rational design decision-making. They permit attributes to be weighted in importance and they support hierarchical aggregation. Thus to answer French’s second question, provided that the axioms of design-appropriateness adequately reflect the decision-maker’s notion of rationality, the method of imprecision can allow the decision-maker to define an aggregation hierarchy that acceptably models how the decision-maker might actually trade-off preferences. The method of imprecision seeks to guide the designer in creating a model decision-maker that accurately reflects the design decision while embodying appropriate canons of rationality for engineering design.

The electric vehicle example in Section 2.7 demonstrated the modeling of a design problem, and in particular the process of identifying design variables, performance variables, and design considerations and constructing the design preference aggregation hierarchy. The elucidation of this process is the second key contribution of this thesis.

The feasibility of the process of identifying design considerations and constructing the design preference aggregation hierarchy was demonstrated for an electric vehicle design in Section 2.7. The process of enumerating design considerations, explicitly relating design considerations to design variables, constructing a hierarchy, determining relative importance and degree of compensation in aggregation, and examining the resulting model, was in itself a valuable exercise. The careful analysis of how the design variables impact design considerations clarified many important issues. Thus in reply to French’s third question, it is suggested that the process of constructing the method of imprecision model is not only feasible and informative, but also requires the designer to more clearly distinguish and explicitly quantify the beliefs and preferences that are to be modeled.

In the method of imprecision, design preferences  $\mu_{d_i}$  are specified on design variables and functional requirements  $\mu_{p_j}$  are specified on performance variables. Individual design preferences  $\mu_{d_i}(d_i)$  are aggregated into the combined design preference  $\mu_d(\vec{d})$ , and individual functional requirements  $\mu_{p_j}(p_j)$  are aggregated into the combined functional requirement  $\mu_p(\vec{p})$ . The aggregation of these combined preferences into the overall preference  $\mu_o$  is complicated by the need to map preferences between the DVS and the PVS:

- The combined design preference must be mapped to  $\mu_d(\vec{p})$  on the PVS to obtain  $\mu_o(\vec{p}) = \mathcal{P}(\mu_d(\vec{p}), \mu_p(\vec{p}))$ .
- The combined functional requirement must be mapped to  $\mu_p(\vec{d})$  onto the DVS to obtain  $\mu_o(\vec{d}) = \mathcal{P}(\mu_d(\vec{d}), \mu_p(\vec{d}))$ .

Previously, design preferences were mapped onto the PVS using the Level Interval Algorithm (LIA). The LIA has four important limitations:

1. The  $\alpha$ -cuts  $D_{\alpha_k}^d$  must be  $n$ -cubes, which is accurate only for aggregation by  $\mathcal{P}_{\min}$ .
2. The performance variable endpoints  $p_{j_{\min}}^{\alpha_k}$  and  $p_{j_{\max}}^{\alpha_k}$  calculated are only correct under certain conditions: in practice these require that  $f_j$  be monotonic.
3. The endpoints obtained only indicate extremal points on  $P_{\alpha_k}^d$ , the combined design preference  $\alpha$ -cut in the PVS: the full geometry of  $P_{\alpha_k}^d$  is not determined.
4. Up to  $2^n$  function evaluations are required to evaluate each  $\alpha$ -cut, a number that quickly becomes prohibitive as the number of design variables  $n$  increases.

The primary limitation of the LIA, that it requires monotonicity, may be removed by reformulating the problem as a constrained optimization.

In order to address the issue of robustness as well as limitations 1 and 3, an approximation  $\vec{f}'$  for  $\vec{f}: \text{DVS} \rightarrow \text{PVS}$  is constructed over  $D_\epsilon^d$  (the  $\alpha$ -cut at infinitesimal  $\alpha = \epsilon$ ). Obtaining a linear approximation  $\vec{f}'$  fulfills four purposes:

1. It removes acceptably linear design variables from the search space for optimization.
2. It supplies a global approximation to  $\vec{f}$  over  $D_\epsilon^d$  for determining the geometry of  $P_{\alpha_k}^d$  between extremal points.
3. It enables the calculation of design sensitivities  $\kappa_{ji}^{\alpha_k}$ .
4. It provides a computationally tractable, albeit approximate, means to map the combined functional requirement from the PVS onto the DVS.

The mapping of  $D_{\alpha_k}^d$  onto the PVS does not depend entirely upon the accuracy of the linear approximation  $\vec{f}^l$ . The shape of  $P_{\alpha_k}^d$  in the PVS is estimated by obtaining extremal points in each  $p_j$  via optimization (facilitated by linear approximation), and then interpolating the bounding edges between points using  $\vec{f}^l$ . The linear approximation  $\vec{f}^l$  furnishes additional information about the shape of  $P_{\alpha_k}^d$  away from extremal points that would otherwise be unavailable.  $\vec{f}^l$  is used to provide approximate information not to replace precise information, but to replace a lack of information.

The linear approximation  $\vec{f}^l$  over  $D_\epsilon^d$  is obtained using techniques adapted from experiment design. The combined use of experiment design to explore the design space and optimization assisted by linear approximation to map preferences is the third key contribution of this thesis. Using experiment design to obtain a linear regression model is efficient in function evaluations, does not require advanced statistical techniques, and is well-suited to computer implementation. Experiment design evaluates a balanced set of points in the search space  $D_\epsilon^d$  in order to characterize  $\vec{f}$  and construct  $\vec{f}^l$ . The likely location of the global minimum is chosen as the starting point for optimization.

The fourth key contribution of this thesis is the provision of a fractional precision that permits the designer to trade-off the number of function evaluations against the quality of the answer obtained. This adjustment allows the designer to use the same computer program to obtain quick estimates as well as precise evaluations. The fractional precision determines not only the tolerance of the optimization algorithm,

but also the necessary conditions for  $\vec{f}'$  to approximate  $\vec{f}$  sufficiently accurately in each design variable  $d_i$ . Only sufficiently linear variables are approximated for optimization. Optimization proceeds on  $\vec{f}$  over the remaining non-linear variables.

The Imprecise Design Tool (IDT) was developed to verify the algorithms described in Chapter 3 and to demonstrate the method of imprecision on engineering problems. Section 4.1 discussed an application of the IDT to the Engine Development Cost Estimator (EDCE) provided by General Electric Aircraft Engines, Cincinnati, Ohio. The EDCE estimates the cost of developing a new aircraft engine and is one of several programs that together estimate the total lifetime cost of an engine.

Section 4.1.1 presented a turbofan aircraft engine development problem which involved two (imprecisely specified) options:

1. Develop the new engine from an existing turbojet design.
2. Modify an existing, but dated, turbofan design.

The IDT was used to map design preferences onto the one-dimensional PVS. The specification of a relative functional requirement for minimizing cost was shown to be problematic.

A more recent application of the IDT, to an automobile body design problem, was presented in Section 4.2. Noise, vibration, and harshness (NVH) design is concerned with three aspects of vibration: audible noise, tactile vibration, and subjective evaluations of safety, comfort, and luxury based on perceived levels of noise and vibration. Measures of static rigidity (bending and torsional stiffness) are indicators of perceived safety, comfort, and luxury levels. Measures of dynamic response directly predict noise and vibration characteristics [12].

At this particular U.S. automobile manufacturer, the static and dynamic response of the body-in-white is calculated using a commercial finite element package. Each finite element calculation of static and dynamic response requires significant supercomputer time: approximately 15 seconds to evaluate a simplified model [12]. Therefore, in applying the method of imprecision to this problem, an important

consideration was to minimize the number of function evaluations incurred. The optimization and experiment design techniques described in Chapter 3 were developed to address this issue.

In Section 4.2.1 the IDT was applied to a finite element model of the passenger compartment of a hypothetical four-door body-in-white. Bending and torsional stiffnesses were calculated using finite element methods. Ten design variables were defined including the gauges of key members and the fore-aft location of the B pillar. This example attempted to demonstrate the feasibility of the methods introduced in Chapter 3.

Comparing the number of function evaluations required for resolution III and IV experiments and fractional precisions of 1 and 0.01 showed that, for this example, the results and the number of function evaluations required for optimization were independent of resolution. Approximation was also shown to be effective in minimizing the number of additional function evaluations required by optimization: for a fractional precision of 1, only two additional function evaluations were required. Moreover, even a resolution III experiment combined with a fractional precision of 1 resulted in calculated extrema that were extremely close to the best extrema found. The apparent effectiveness of the methods developed poses an intriguing question: if it is possible to obtain the correct answer with a resolution III experiment and a fractional precision of 1 requiring only 44 function evaluations, why would additional function evaluations be necessary? It was suggested in Chapter 3 that a resolution IV experiment buys additional reliability and confidence in the results, and a smaller fractional precision limits the allowable inaccuracy. Yet this example appears to show that the central issue is not accuracy but confidence in the results. Additional function evaluations may find better extrema. A resolution IV experiment combined with a smaller fractional precision, however, is less likely to miss the correct extrema.

These results demonstrate that a selectively applied linear approximation can be surprisingly effective in evaluating an example finite element model of an automobile body. Finite element models are widely used in industry. The automobile body

design and aircraft engine development examples represent two contrasting design problems from industry. The verification of the algorithms for method of imprecision calculations on realistic design problems is the fifth key contribution of this thesis.

In Section 4.2.2, the methods introduced in Chapter 3 for mapping  $\alpha$ -cuts from the DVS to the PVS were demonstrated on quadratic approximations to bending and torsional stiffness. Despite the pronounced non-linearity of bending stiffness, the correct extrema were found when a fractional precision of 0.01 was specified. However, the geometry of  $P_\epsilon^d$ , the combined design preference  $\alpha$ -cut  $D_\epsilon^d$  mapped onto the PVS, was poorly approximated by  $P_\epsilon^{d'}$ . Where  $\vec{f}$  is sufficiently non-linear that  $P_\epsilon^{d'}$  obtained via linear approximation is inadequate, effective methods to more accurately determine the geometry of  $P_\epsilon^d$  have yet to be developed within the method of imprecision. Developing these methods is an important goal for future research.

The specification of design preferences  $\mu_{d_i}$  and functional requirements  $\mu_{p_j}$  is significantly more difficult than defining what they represent. Moreover, the form of these specified preferences is dependent on the algorithms used. The algorithms described in Chapter 3 for mapping the combined design preference  $\mu_d$  from the DVS to the PVS rely on approximating  $\mu_d(\vec{d})$  as  $\alpha$ -cut sets  $D_{\alpha_1}^d, \dots, D_{\alpha_{M_I}}^d$ .

Aggregating discretized preference functions with an aggregation function other than  $\mathcal{P}_{\min}$  creates additional non- $n$ -cubic level sets (Section 3.3). Although it might seem that finer discretization would lead to more accurate calculations, there are two compelling reasons why fewer  $\alpha$ -cuts may be better.

1. Each additional level set defined for the individual design preferences potentially adds not just one new level set for the combined design preference, but as many new level sets as there are new combinations of  $\alpha$ . Each new level set incurs additional function evaluations.
2. The accuracy with which design preferences are represented should not exceed the accuracy with which they can be elicited, or the accuracy with which the results can be interpreted.

The two most important  $\alpha$ -cut intervals are at  $\alpha = 1$ , corresponding to ideal

variable values, and  $\alpha = \epsilon$  ( $0 < \epsilon \ll 1$ ), corresponding to barely acceptable variable values. The  $\alpha$ -cut interval at  $\alpha = 1$  identifies the ideal or target range of values for the variable. The  $\alpha$ -cut interval at  $\alpha = \epsilon$  identifies the largest acceptable range of values for the variable. Aggregating a preference of  $\epsilon$  with any other preference values results in an infinitesimal preference which is effectively equal to  $\epsilon$ . Aggregation also cannot create new preference levels between  $\alpha_1 = \epsilon$  and the second lowest  $\alpha$ -cut preference  $\alpha_2$ . Thus defining an  $\alpha$ -cut at  $\alpha = \epsilon$  is computationally efficient.

A basic implementation of the method of imprecision suitable for preliminary analyses would represent both design preferences and functional requirements as acceptable and ideal intervals, thus reducing the procedure to the propagation and intersection of acceptable and ideal intervals. It is unnecessary to identify aggregation functions or even an aggregation hierarchy. Relatively few function evaluations are required. Moreover, it is easier to specify a set of ideal and acceptable intervals than a continuous functional requirement, and the results are more easily interpreted.

A more complete implementation would use three  $\alpha$ -cuts:  $\alpha_1 = \epsilon$  (undesirable yet barely acceptable),  $\alpha_2 = 0.5$  (neutral or indifferent), and  $\alpha_3 = 1$  (ideal or target). Up to  $n$  new preference levels are created between  $\alpha_2$  and  $\alpha_3$ . These intermediate levels allow different aggregation functions to be modeled: they encode trade-offs between design variables that are ignored in the basic two  $\alpha$ -cut implementation. The third  $\alpha$ -cut is necessary to fully implement the method of imprecision.

For group decision-making, specifying preferences and determining a preference aggregation hierarchy are subject to bargaining or negotiation. One suggested solution is to appoint a facilitator, who would weigh the opinions of the members of the design team in order to construct the method of imprecision model. Whenever preferences or hierarchies need to be modified the facilitator should be required to choose only one participant's opinions for revision. For large groups, several participants could be chosen for each iteration. This provides an incentive to submit information that is realistic instead of exaggerated, as well as a mechanism for building consensus.

A theory for representing and evaluating design precision is incomplete without feasible and efficient computational algorithms. Computational algorithms must be verified on realistic problems. Yet even a methodology complete with efficient computational algorithms that have been verified on design problems from industry is still incomplete without practical procedures for implementation. The procedures described in Section 5.1, which are an essential step towards bringing the method of imprecision to design problems in industry, are the sixth key contribution of this thesis. These procedures reduce the abstract mathematics of representing preference to readily understood acceptable, neutral, and ideal intervals and allow for both preliminary and detailed analyses. Moreover, through an informal arbitration procedure for determining preferences and aggregation hierarchies, the difficult but ever-present problem of bargaining can be addressed. An effective means to control bargaining behavior is a prerequisite for applying the method of imprecision to support group design decisions.

In Section 5.2 the electric vehicle design problem from Section 2.7 was used to present a possible scenario for implementing the method of imprecision. The scenario involved the development of a new variant in a family of electric vehicles based on a common basic chassis. The full treatment of the electric vehicle example in Sections 2.7 and 5.2 addresses French's first question by demonstrating that an aggregation hierarchy of explicitly defined design and performance considerations is capable of modeling imprecision for a realistic design problem. The example is intended to reflect the detail that is often difficult to model formally, but is an intrinsic part of real design problems. The flexibility of the model decision problem postulated by the method of imprecision enables these details to be represented in a formal decision-making methodology.

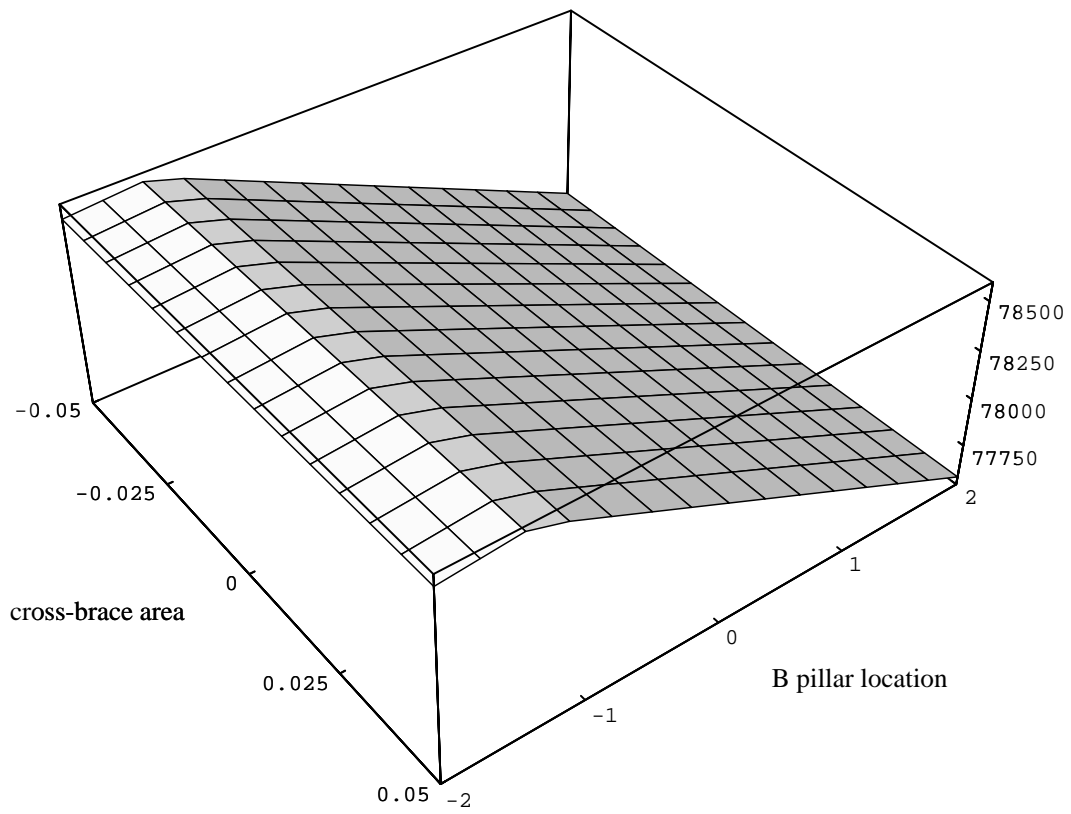
## Appendix A

### Approximation by Tiling

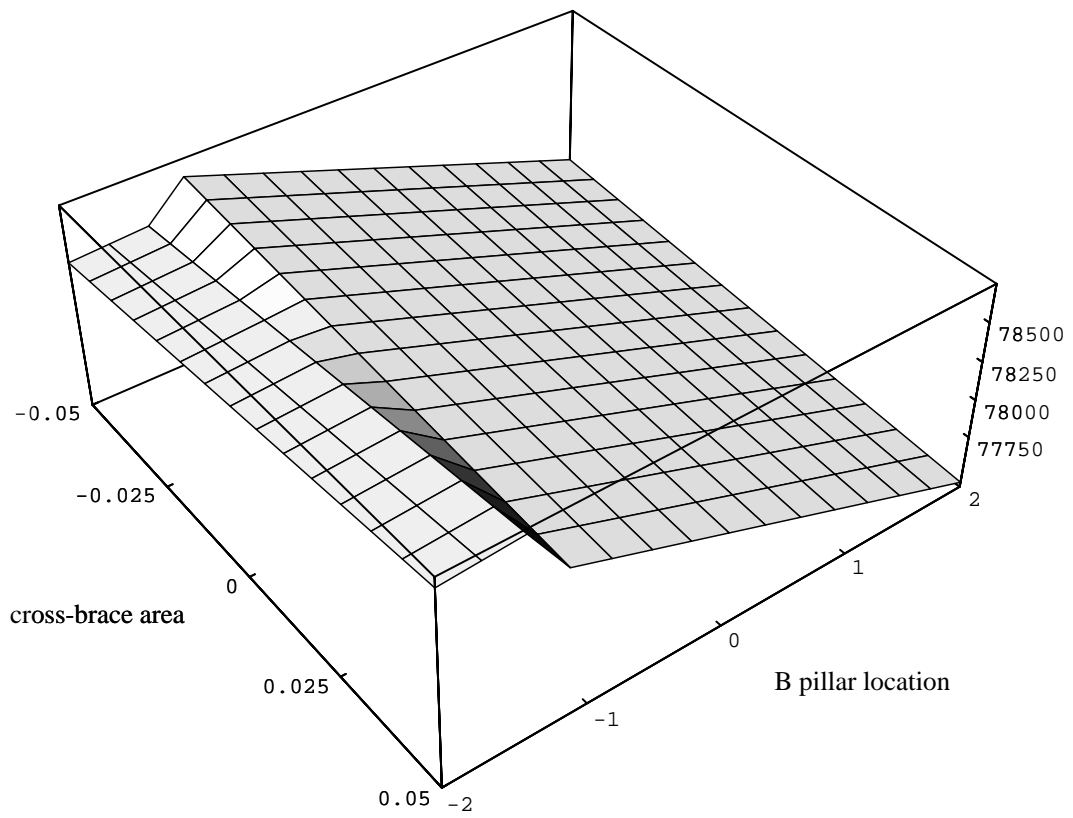
This section presents an alternative method for approximating  $\vec{f} : \text{DVS} \rightarrow \text{PVS}$  that was ultimately shown to be impractical. The quadratic example presented in Section 4.2 highlights a basic deficiency in approximating  $\vec{f}$  as a linear mapping. Yet a quadratic approximation introduces curved surfaces that would be difficult to calculate and to interpret. An alternative is a piecewise linear approximation.

The idea was to successively partition  $D_\epsilon^d$  into  $n$ -cubic tiles. A design variable  $d_i$  that was not acceptably linear would be approximated by multiple tiles in the  $d_i$  direction. A linear approximation would be constructed on each tile. This is a type of branch-and-bound scheme [59]. The quadratic bending stiffness shown in Figure 4.11 could be approximated by two such tiles, as shown in Figure A.1. Tiling preserves some of the simplicity of linear approximations while allowing the function to exhibit internal extrema.

Figure A.1 does not show the many potential problems with constructing a tiled linear approximation. The  $K_B$  surface is not significantly skewed or twisted. For less well behaved functions, linear approximations on adjacent tiles may be misaligned, leading to discontinuities near each end of the shared edge (Figure A.2). Extrema will in general not be aligned with any particular variable direction and thus the boundary between two  $n$ -cubic tiles will not accurately follow the ridge or trough in the function. Tiles should be  $n$ -cubes in order to simplify the construction of linear approximations on each tile using experiment design, and also because allowing



**Figure A.1** Tiling approximation for  $K_B$ .



**Figure A.2** Misaligned tiles.

sloping boundaries would introduce significant complexity into the algorithm. If the linear approximations are not to be constructed using  $n$ -cubic experiment designs, then this restriction may be relaxed, though convenient methods for constructing and representing such a tiling and for using it to map  $\alpha$ -cuts may not exist.

The disadvantages of tiling the search space become clearer as the number of design variables grows large. Each partitioning of  $D_\epsilon^d$  requires the evaluation of an experiment design for each of the two new tiles created. Only half of the previously evaluated points can be reused. Thus each partitioning adds one complete experiment to the total number of points evaluated. For two design variables, one central composite design requires only nine evaluations. Four tiles that partition  $D_\epsilon^d$  in both variables require a total of 36 evaluations. For eight design variables, each resolution III central composite design requires 29 evaluations. Sixteen tiles that partition  $D_\epsilon^d$  in four of the eight variables requires a total of 464 evaluations. As  $n$  increases, the number of points evaluated in each experiment, the number of potentially non-linear variables, and hence the number of necessary partitions, all increase. A test implementation of tiling expended a rapidly increasing number of function evaluations as the fractional precision was refined. Moreover, it was difficult to determine whether, for a given fractional precision, the algorithm could complete its calculation in any reasonable length of time: the number of function evaluations required was impossible to predict.

## Bibliography

- [1] J. Aczél. *Lectures on Functional Equations and their Applications*. Academic Press, New York, 1966.
- [2] P. Adby and M. Dempster. *Introduction to Optimization Methods*. Chapman and Hall, London, 1974.
- [3] J. Alger and C. Hays. *Creative Synthesis in Design*. Prentice Hall, Englewood Cliffs, 1964.
- [4] T. Barker. *Quality by Experimental Design*. Marcel Dekker, Inc., New York, 1985.
- [5] D. L. Bartel and R. W. Marks. The optimum design of mechanical systems with competing objectives. *Journal of Engineering for Industry*, pages 171–178, February 1974.
- [6] R. E. Bellman and L. A. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 17(4):B-141–163, December 1970.
- [7] D. M. Byrne and S. Taguchi. The Taguchi approach to parameter design. In *Quality Congress Transaction – Anaheim*, pages 168–177. ASQC, May 1986.
- [8] H. W. Chi and C. L. Bloebaum. Mixed variable optimization using Taguchi's orthogonal arrays. In *Advances in Design Automation - 1995*, volume DE-82, pages 501–508, New York, September 1995. ASME.

- [9] Kim B. Clark and Takahiro Fujimoto. *Product Development Performance: Strategy, Organization and Management in the World Auto Industry*. Harvard Business School Press, Boston, MA, 1991.
- [10] J. Corbett. Design for economic manufacture. *Annals of the C.I.R.P.*, 35(1):93–97, 1986.
- [11] Discussions with engineers at the Alcoa Technical Center, Monroeville, Pennsylvania, December 11–12, 1995.
- [12] Discussions with engineers at a major U.S. automobile manufacturer, November 7–11, 1994.
- [13] Discussions with managers and engineers at General Electric Corporate Research and Development, Schenectady, New York, January 10, 1994.
- [14] D. C. Dlesk and J. S. Liebman. Multiple objective engineering optimization. *Engineering Optimization*, 6:161–175, 1983.
- [15] W. M. Dong and F. S. Wong. Fuzzy weighted averages and implementation of the extension principle. *Fuzzy Sets and Systems*, 21(2):183–199, February 1987.
- [16] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [17] Simon French. Fuzzy decision analysis: Some criticisms. In *TIMS/Studies in the Management Sciences*, pages 29–44. Elsevier Science Publishers B.V., North-Holland, 1984.
- [18] L. Fung and K. Fu. An axiomatic approach to rational decision making in a fuzzy environment. In L. Zadeh et al., editors, *Fuzzy Sets and Their Applications to Cognitive and Design Processes*, New York, 1974. Academic Press.
- [19] A.M. Geoffrion, J.S. Dyer, and A. Feinberg. An interactive approach for multicriterion optimization, with an application to the operation of an academic department. *Management Science*, 19(4):357–368, 1972.

- [20] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts, 1989.
- [21] R. H. Hardin and N. J. A. Sloane. Codes (spherical) and designs (experimental). *Proceedings of Symposia in Applied Mathematics*, 50:179–206, 1995.
- [22] E. B. Haugen. *Probabilistic Mechanical Design*. John Wiley and Sons, New York, 1980.
- [23] <http://www.calstart.org/about/programs/p-chass.html>.
- [24] A. Kaufmann and M. M. Gupta. *Fuzzy Mathematical Models in Engineering and Management Science*. North Holland, Amsterdam, 1988.
- [25] R. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York, 1976.
- [26] H. B. Keller. *Lectures on Numerical Methods in Bifurcation Problems*. Springer-Verlag, Berlin, 1987.
- [27] G. J. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- [28] J. C. Korngold and G. A. Gabriele. Multidisciplinary analysis and optimization of discrete problems using response surface methods. In *Advances in Design Automation - 1995*, volume DE-82, pages 173–180, New York, September 1995. ASME.
- [29] William S. Law and Erik K. Antonsson. Implementing the Method of Imprecision: An Engineering Design Example. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '94)*, volume 1, pages 358–363. IEEE, June 1994. Invited paper.
- [30] William S. Law and Erik K. Antonsson. Including Imprecision in Engineering Design Calculations. In *Design Theory and Methodology - DTM '94*, volume DE-68, pages 109–114. ASME, September 1994.

- [31] William S. Law and Erik K. Antonsson. Hierarchical Imprecise Design with Weights. In *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE/IFES'95)*, volume 1, pages 383–388. IEEE, March 1995.
- [32] William S. Law and Erik K. Antonsson. Optimization Methods for Calculating Design Imprecision. In *Advances in Design Automation - 1995*, volume 1, pages 471–476. ASME, September 1995.
- [33] William S. Law and Erik K. Antonsson. Multi-dimensional Mapping of Design Imprecision. In *10th International Conference on Design Theory and Methodology*. ASME, 1996. Submitted for review, Jan. 10, 1996.
- [34] T. Mathai and D. Cronin. Application of fuzzy sets to car body structure NVH design. *Proceedings of IX International Conference on Vehicle Structural Mechanics and CAE*, pages 107–115, April 1995.
- [35] J. D. Mattingly, W. H. Heiser, and D. H. Daley. *Aircraft Engine Design*. AIAA Education Series. AIAA, New York, 1987.
- [36] K. Menger. Statistical metrics. *Proceedings of the National Academy of Sciences*, 28:535–537, 1942.
- [37] N. F. Michelena and P. Y. Papalambros. Optimal model-based decomposition of powertrain system design. In *Advances in Design Automation - 1995*, volume DE-82, pages 165–172, New York, September 1995. ASME.
- [38] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, New York, 1991.
- [39] K. Müller and M. Thäringen. Applications of fuzzy hierarchies and fuzzy MADM methods to innovative system design. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, volume 1, pages 364–367. IEEE, June 1994.
- [40] C. V. Negoită and D. A. Ralescu. *Applications of Fuzzy Sets to Systems Analysis*. Halsted Press, New York, 1975.

- [41] Vilém Novák. *Fuzzy Sets and Their Applications*. Adam Hilger, Philadelphia, 1989. Published in English by IOP Publishing Ltd.
- [42] L. O'Connor. Electric vehicles move closer to market. *Mechanical Engineering*, 117(3):139–152, March 1995.
- [43] Kevin N. Otto. *A Formal Representational Theory for Engineering Design*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, June 1992.
- [44] Kevin N. Otto and Erik K. Antonsson. Trade-Off Strategies in Engineering Design. *Research in Engineering Design*, 3(2):87–104, 1991.
- [45] Kevin N. Otto and Erik K. Antonsson. Trade-Off Strategies in the Solution of Imprecise Design Problems. In T. Terano et al., editors, *Fuzzy Engineering toward Human Friendly Systems: Proceedings of the International Fuzzy Engineering Symposium '91, Volume 1*, pages 422–433, Yokohama, Japan, November 1991. LIFE, IFES.
- [46] Kevin N. Otto, Andrew D. Lewis, and Erik K. Antonsson. Approximating  $\alpha$ -cuts with the Vertex Method. *Fuzzy Sets and Systems*, 55(1):43–50, April 1993.
- [47] Kevin N. Otto, Andrew D. Lewis, and Erik K. Antonsson. Determining optimal points of membership with dependent variables. *Fuzzy Sets and Systems*, 60(1):19–24, November 1993.
- [48] G. Pahl and W. Beitz. *Engineering Design*. The Design Council, Springer-Verlag, New York, 1984.
- [49] P. Papalambros and D. Wilde. *Principles of Optimal Design*. Cambridge University Press, New York, 1988.
- [50] G. S. Peace. *Taguchi Methods: a Hands-On Approach*. Addison-Wesley, Reading, MA, 1993.
- [51] M. Phadke. *Quality Engineering Using Robust Design*. Prentice Hall, Englewood Cliffs, NJ, 1989.

- [52] S. Pugh. *Total Design*. Addison-Wesley, New York, 1990.
- [53] Thomas L. Saaty. Axiomatic foundation of the analytic hierarchy process. *Management Science*, 32:841–855, 1986.
- [54] Michael J. Scott and Erik K. Antonsson. Aggregation Functions for Engineering Design Trade-offs. *Fuzzy Sets and Systems*, 1996. Submitted for review, July, 1995.
- [55] Michael J. Scott and Erik K. Antonsson. Formalisms for Negotiation in Engineering Design. In *10th International Conference on Design Theory and Methodology*. ASME, 1996. Submitted for review, January 10, 1996.
- [56] J. N. Siddall. *Probabilistic Engineering Design; Principles and Applications*. Marcel Dekker, New York, 1983.
- [57] G. Taguchi and M. Phadke. Quality engineering through design optimization. *IEEE*, pages 1106–1113, 1984.
- [58] C. H. Tseng and T. W. Liu. Mimimax multiobjective optimization in structural design. *International Journal for Numerical Methods in Engineering*, 30:1213–1228, 1990.
- [59] H. Tuy and R. Horst. Convergence and restart in branch-and-bound algorithms for global optimization. application to concave minimization and d.c. optimization problems. *Mathematical Programming*, 41:161–183, 1988.
- [60] Karl T. Ulrich and Steven D. Eppinger. *Product Design and Development*. McGraw-Hill, New York, 1995.
- [61] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, second edition, 1953.
- [62] A. C. Ward and W. P. Seering. The performance of a mechanical design compiler. memo 1084, MIT Artificial Intelligence Laboratory, 1989.

- [63] A. C. Ward and W. P. Seering. Quantitative inference in a mechanical design compiler. memo 1062, MIT Artificial Intelligence Laboratory, 1989.
- [64] Allen C. Ward, Jeffrey K. Liker, Durward K. Sobek, and John J. Cristiano. Set-based concurrent engineering and Toyota. In *Design Theory and Methodology – DTM '94*, volume DE68, pages 79–90. ASME, September 1994.
- [65] Allen C. Ward, Jeffrey K. Liker, Durward K. Sobek, and John J. Cristiano. The second Toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Review*, 36(3):43–61, Spring 1995.
- [66] Kristin L. Wood. *A Method for Representing and Manipulating Uncertainties in Preliminary Engineering Design*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1989.
- [67] Kristin L. Wood and Erik K. Antonsson. Computations with Imprecise Parameters in Engineering Design: Background and Theory. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 111(4):616–625, December 1989.
- [68] Kristin L. Wood and Erik K. Antonsson. Modeling Imprecision and Uncertainty in Preliminary Engineering Design. *Mechanism and Machine Theory*, 25(3):305–324, February 1990. Invited paper.
- [69] Kristin L. Wood, Kevin N. Otto, and Erik K. Antonsson. Engineering Design Calculations with Fuzzy Parameters. *Fuzzy Sets and Systems*, 52(1):1–20, November 1992.
- [70] R. Yager. Fuzzy decision making including unequal objectives. *Fuzzy Sets and Systems*, 1:87–95, 1978.
- [71] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [72] H. J. Zimmermann. *Fuzzy Set Theory - and Its Applications*. Management Science/Operations Research. Kluwer-Nijhoff Publishing, Boston, MA, 1985.

- [73] H.-J. Zimmermann and H.-J. Sebastian. Fuzzy design - Integration of fuzzy theory with knowledge-based system-design. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, volume 1, pages 352–357. IEEE, June 1994.
  
- [74] H.-J. Zimmermann and H.-J. Sebastian. Intelligent system design support by fuzzy-multi-criteria decision making and/or evolutionary algorithms. In *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE/IFES'95)*, volume 1, pages 367–374. IEEE, March 1995.

## Index

- $\alpha$ -cut, 48, 78, 107
  - $n$ -cubic, 52, 114
- absolute
  - functional
    - requirement, 87
  - preference, 35
- acceptable interval, 61, 116, 122
- acceptably linear, 66, 96
- aggregation, 12
  - $\alpha$ -cuts, 52
  - degree of
    - compensation, 22, 30, 126
  - design-appropriate, 15, 17
  - fully compensating, 16, 126
  - function, 12
  - hierarchical, 23, 28
  - hierarchy, 28, 81, 125, 128
  - maximally
    - compensating, 21, 126
  - non-compensating, 15, 78, 126
- operation, 23
  - partially
    - compensating, 20
  - supercompensating, 21
  - weighted means, 19
  - weights, 17, 23, 25
- AHP, 36
- annihilation, 14, 18, 25
- approximation
  - linear, 56
  - piecewise linear, 145
  - quadratic, 103
  - tiling, 145
- axioms
  - annihilation, 14, 18, 25
  - commutativity, 13, 17
  - continuity, 14, 18
  - idempotency, 14, 18
  - monotonicity, 13, 18
  - self-normalization, 19, 25
- backward mapping, 58, 124, 129
- bargaining, 123
- basic implementation, 117, 122
- bending stiffness, 95, 103
- body-in-white, 91
- branch-and-bound, 145
- central composite
  - design, 68
- combined
  - design preference, 47, 55, 79, 87
  - functional
    - requirement, 47, 58, 87
- commutativity, 13, 17
- compensation, degree of, 22, 30, 126
- concept selection
  - matrices, 36
- confounding, 64
- continuation methods, 56
- continuity, 14, 18
- degree of compensation, 22, 30, 126

- design considerations,
  - 12, 29, 81
- design preference, 11,
  - 81, 122
  - n*-cubic  $\alpha$ -cut, 52
  - combined, 47, 55, 79, 87
  - interval, 48, 78
  - mapping, 52
- design sensitivity, 57, 98,
  - 99
- design teams, 123
- design variable, 8, 29,
  - 76, 95, 121
  - discrete, 51, 78
- DOE, 63, 92
- DVS, 7
- EDCE, 76
- electric vehicle, 26, 121
- error
  - linear approximation, 69, 97, 106
- extension principle, 47
- facilitator, 123
- forward mapping, 52,
  - 122, 129
- fractional factorial, 64
- fractional precision, 51,
  - 65, 94, 100, 103
- French's 3 questions, 4
- function evaluations, 64,
  - 65, 68, 79, 90, 92, 101, 115, 148
- functional requirement,
  - 10, 81, 122
  - absolute, 87
  - combined, 47, 58, 87
  - interval, 49, 78
  - mapping, 58
- fuzzy sets, 3, 41
- FWA, 48
- group decision-making,
  - 123
- ideal interval, 61, 116,
  - 122, 126
- idempotency, 14, 18
- IDT, 76, 90
- implementation
  - basic, 117, 122
  - three  $\alpha$ -cut, 118, 125
- imprecision, 1, 9, 33
- infinitesimal preference,
  - 55
- interval
  - acceptable, 61, 116, 122
  - design preference, 48, 78
  - functional requirement, 49, 78
  - ideal, 61, 116, 122, 126
  - neutral, 118, 126
- LIA, 48, 79
- linear approximation, 56
  - error, 69, 97, 106
- lookup table, 79
- mapping, 78
  - backward, 58, 124, 129
  - design preference, 52
  - forward, 52, 122, 129
  - functional requirement, 58
- method of imprecision,
  - 3, 42
  - implementation, 117, 118
  - previous work, 4
  - steps in, 132
- monotonicity, 13, 18, 50,
  - 51, 66
- negotiation, 123
- neutral interval, 118, 126
- NVH design, 90, 122
- offset, 56, 68, 99
- one at a time search, 50,
  - 51
- one-factor-at-a-time, 66
- optimization, 37, 50, 57,
  - 90, 92, 100, 106
- orthogonal arrays, 63
- overall preference, 12,
  - 47, 87, 128
- performance
  - considerations, 10,

- 122
- performance variable, 9,  
28, 78, 121
- Powell's method, 50
- precision, fractional, 51,  
65, 94, 100, 103
- preference, 9
- n*-cubic  $\alpha$ -cut, 114
- absolute, 35
- design, 11
- functional
- requirement, 10
- infinitesimal, 55
- intermediate, 52, 114
- overall, 12, 47, 87,  
128
- probability, 33
- Pugh charts, 36
- PVS, 9
- quadratic
- approximation, 103
- resolution, 64, 97
- RFP, 80
- robustness, 50
- running chassis, 26
- self-normalization, 19,  
25
- set-based design, 1, 38
- stiffness
- bending, 95, 103
- torsional, 95, 98, 103
- Taguchi's method, 33
- three  $\alpha$ -cut
- implementation,  
118, 125
- tiling approximation,  
145
- torsional stiffness, 95,  
98, 103
- trade-off
- fully compensating,  
16, 126
- maximally
- compensating, 21,  
126
- non-compensating,  
15, 78, 126
- partially
- compensating, 20
- precision vs.
- evaluations, 51, 65,  
94, 102
- strategy, 12
- supercompensating,  
21
- turbofan, 80
- utility theory, 34
- Vertex Method, 48
- weighted means, 19
- weights, 17, 25, 30, 126