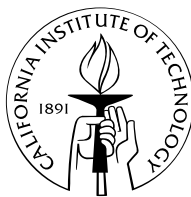


Set Mapping in the Method of Imprecision

Thesis by
Xiaou Wang

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



California Institute of Technology
Pasadena, California

2003

(Defended Sept. 26, 2002)

© 2003

Xiaoou Wang

All rights Reserved

Acknowledgements

First I would like to thank my advisor, Professor Erik Antonsson, for his advice and encouragement during the last five years. Almost all my ideas were inspired by the discussions with him. I can not finish this work without his help.

And I also would like to say thanks to some other students at Caltech. I really appreciate the help offered by William S. Law and Michael J. Scott when they already left Caltech. The hardware testing and the solid model were works done by two undergraduates, Zhee Khoo and Juan Nuño.

Finally, I thank Lu and my parents for their support and understanding during my study at Caltech.

Set Mapping in the Method of Imprecision

by

Xiaoou Wang

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

Abstract

The *Method of Imprecision*, or M_0I , is a semi-automated set-based approach which uses mathematics of fuzzy sets to aid the designer making decisions with imprecise information in the preliminary design stage.

The *Method of Imprecision* uses preference to represent the imprecision in engineering design. The preferences are specified both in the design variable space (DVS) and the performance variable space (PVS). To reach the overall preference which is needed to evaluate designs, the mapping between the DVS and the PVS should be explored. Many engineering design tools can only produce precise results with precise specifications, and usually the cost is high. In the preliminary stage, the specifications are imprecise and resources are limited. Hence, it is not cost-effective nor necessary to use these engineering design tools directly to study the mapping between the DVS and the PVS. An interpolation model is introduced to the M_0I to construct metamodels for the actual mapping function between the DVS and the PVS. Due to the nature of engineering design, multistage metamodels are needed. Experimental design is used to choose design points for the first metamodel. In order to find an efficient way to choose design points when *a priori* information is available, many sampling criteria are discussed and tested on two specific examples. The difference between different sampling criteria when the number of added design points is small, while more design points do improve the accuracy of the metamodel substantially.

The metamodels can be used to induce preferences in the DVS or the PVS according to the extension principle. The Level Interval Algorithm (LIA) is a discrete approximate implementation of the extension principle. The resulting preference by the LIA is presented as an α -cut, which is the set of designs or performances with a certain level of preference. There are some limitations of the LIA, especially for multidimensional DVS and PVS. A new extension of the LIA is proposed to

compute α -cuts with more accuracy and less limitations. The designers have more control over the trade-off between the cost and accuracy of the computation with the new extension of the LIA.

The results of the *Method of Imprecision* should be the set of alternative designs in the DVS at a certain preference level, and the set of achievable performances in the PVS. The information about preferences in the DVS and the PVS is needed to transfer back and forth. Usually the mapping from the PVS to the DVS is unavailable, while it is needed to induce preference in the DVS from the PVS. A new method is constructed to compute the α -cuts in both spaces from preferences specified in the DVS and the PVS.

Finally, a new measure is proposed to find the most cost-effective sampling region of new design points for a metamodel. Also, the full implementation of the *Method of Imprecision* is listed in detail. Then it is applied to an example of the structure design of a passenger vehicle, and comparisons are made between the new results and previous results.

Contents

1	Introduction	1
1.1	Organization of Thesis	3
2	The Basics of the Method of Imprecision	6
2.1	The Basic Concepts	6
2.2	The Aggregation Functions	7
2.3	Weights of Preferences	10
2.4	Weighted Means	11
2.5	Hierarchical Aggregation	13
2.6	Summary	14
3	Metamodels for the Mapping between DVS and PVS	15
3.1	Approximation or Interpolation	16
3.2	Interpolation Model	18
3.3	Experimental Design	20
3.4	Base Functions	22
3.4.1	Polynomial Models	22
3.4.2	Nonlinear Regression MARS Model	23
3.5	Method for Selection of the Base Functions	24
3.5.1	Problem Description	24
3.5.2	Test Results	26
3.5.3	Base Functions Selected	31
3.6	Criteria for Sampling Design Points with <i>a priori</i> Information	38
3.7	Tests of Improvement between Metamodels	40
3.7.1	Test with Assumption about the Distribution of the Error	41

3.7.2	Test without Assumption about the Distribution of the Error	42
3.8	First Examples and Results	45
3.9	Second Examples and Results	49
3.10	Discussions and Conclusions	59
4	Computation of Preference in DVS and PVS	68
4.1	The Extension Principle and Level Interval Algorithm (or Vertex Method)	69
4.2	Limitation of original LIA for the Mapping between DVS and PVS	70
4.2.1	Anomalies in the LIA for a Single Preference Function	73
4.2.2	Limitations of LIA for Multiple Design Preferences	76
4.2.3	Limitations of the LIA for Multiple Performance Variables	76
4.3	The Revised LIA	79
4.4	The Computation of the Overall Preference	87
4.5	Summary	96
5	Implementation of the M_OI and Example	98
5.1	Implementation of the M_OI	98
5.1.1	The Difference in the Volumes of α -cuts	98
5.1.2	Implementation Process	100
5.2	Problem Description	100
5.2.1	Design Variables and Performance Variables	101
5.2.2	Design Preferences and Functional Requirements	103
5.2.3	Aggregation of Preferences	106
5.3	Results	108
5.3.1	Computational Cost	112
5.4	Discussion	115
5.5	Summary	115
6	Conclusion	117

List of Figures

1.1	The rectangle in the plane of x_9 and x_{10}	3
1.2	The mapping of the rectangle in the plane of K_B and K_T	4
3.1	Two approximations of one set of sample data.	17
3.2	Geometric model of body-in-white in SDRC I-DEAS.	25
3.3	Finite element model of body-in-white.	26
3.4	Bending stiffness data projected onto 2 dimensions.	28
3.5	(ERMSE/response range) of polynomial and MARS models with 19 design points.	29
3.6	(ERMSE/response range) of polynomial and MARS models with 243 design points.	29
3.7	(ERMSE/response range) of polynomial and MARS models with 3125 design points.	31
3.8	(ERMSE/response range) at all design points.	32
3.9	(ERMSE/response range) at all test points.	33
3.10	(maximum error/response range) of all design points and test points.	34
3.11	(error/response range) at maximum response.	36
3.12	Histogram with fitted normal density of e_1 at 10,000 random test points.	43
3.13	Histogram with fitted normal density of e_2 at 10,000 random test points.	44
3.14	Ratio of ERMSE to response range of the 10-D function.	47
3.15	Ratio of maximum error to response range of the 10-D function.	48
3.16	Histogram with fitted Normal density for the errors of \hat{Y}_1 of the 10-D function.	49
3.17	Normal probability plot for the errors of \hat{Y}_1 of the 10-D function.	50
3.18	Histogram with fitted Normal density for the errors of \hat{Y}_2 of the 10-D function.	51
3.19	Normal probability plot for the errors of \hat{Y}_2 of the 10-D function.	52
3.20	Histogram with fitted Normal density for the errors of \hat{Y}_3 of the 10-D function.	53
3.21	Normal probability plot for the errors of \hat{Y}_3 of the 10-D function.	54
3.22	Ratio of ERMSE to response range of the VW model.	57

3.23	Ratio of maximum error to response range of the VW model.	58
3.24	Histogram with fitted Normal density for the errors of \hat{Y}_1 of the VW model.	59
3.25	Normal probability plot for the errors of \hat{Y}_1 of the VW model.	60
3.26	Histogram with fitted Normal density for the errors of \hat{Y}_2 of the VW model.	61
3.27	Normal probability plot for the errors of \hat{Y}_2 of the VW model.	62
3.28	Histogram with fitted Normal density for the errors of \hat{Y}_3 of the VW model.	63
3.29	Normal probability plot for the errors of \hat{Y}_3 of the VW model.	64
4.1	The preference function of the design variable d_1 or d_2	71
4.2	Example result of LIA.	72
4.3	The performance function $p = f(d)$	74
4.4	$\mu_d(p)$ for μ_d as in Figure 4.1, where $p = 3d^3 + 2.5d$	75
4.5	The combined design preferences of two design variables by different \mathcal{P} s.	77
4.6	$P_{\alpha_k}^d$'s in a 2-D PVS from a 2-D DVS.	78
4.7	$P_{\varepsilon}^{d\sharp}$ and $P_{\varepsilon}^{d\natural}$ in a 2-D PVS from a 2-D DVS.	81
4.8	$D_{0.5}^d(10)$'s by different aggregation functions.	85
4.9	$P_{\varepsilon}^{d\Box}(\mathcal{S}, \mathcal{U})$ with different values of T and U	86
4.10	$\mu_{d_1}(d_1)$, $\mu_{d_2}(d_2)$, $\mu_{p_1}(p_1)$ and $\mu_{p_2}(p_2)$	90
4.11	The shape of $\mu_o(\vec{d})$	91
4.12	$D_{0.5}^{o\Box}(\mathcal{S})$ for different values of \mathcal{S}	92
4.13	$P_{0.5}^{o\Box}(\mathcal{S}, \mathcal{U})$ for different values of \mathcal{S} and \mathcal{U}	93
4.14	$P_{0.5}^{d\Box}(8, 16)$ and $P_{0.5}^{o\Box}(8, 16)$ by the forward calculation.	94
4.15	$P_{0.5}^{o\Box}(8, 16)$ by the new method and $P_{0.5}^{o\Box}(1, 1)$ and $P_{0.5}^{o\Box}(8, 16)$ by the forward calculation.	95
5.1	Testing setup of body-in-white.	101
5.2	Geometric model of body-in-white in SDRC I-DEAS.	102
5.3	Finite element model of body-in-white.	102
5.4	Design preferences of the VW model.	104
5.5	Functional requirements of the VW model.	105
5.6	Aggregation hierarchy of preferences.	106
5.7	The α -cuts of design variables at \vec{d}^*	109
5.8	The cross sections of $D_{\alpha_k^{o\Box}}$ at \vec{d}^*	111

5.9	The α -cuts of performance variables at \vec{p}^*	113
5.10	The cross sections of $P_{\alpha_k^{\rho\Box}}$ at \vec{p}^*	114

List of Tables

3.1	Data Fitting Models.	27
3.2	Selected Numerical Results.	30
3.3	Error at Maximum Response (3,364.9).	35
3.4	Ratio of ERMSE to Response Range of the 10-D Function.	46
3.5	Ratio of Maximum Error to Response Range of the 10-D Function.	47
3.6	Significance Levels of Different Tests on the 10-D Function.	50
3.7	Ratio of ERMSE to Response Range of the VW Model.	56
3.8	Ratio of Maximum Error to Response Range of the VW Model.	56
3.9	Significance Levels of Different Tests on the VW Model.	60
5.1	Range of the Design Variables.	108

Chapter 1

Introduction

The *Method of Imprecision*, or M_OI , is a *semi-automated* set-based approach which uses the mathematics of fuzzy sets to aid the designer making decisions with imprecise information in the preliminary design stage [62, 28].

In the preliminary design stage, *imprecision* is the design engineer's uncertainty in choosing among alternatives, and it arises primarily from choices not yet made because of the intrinsic vagueness in the design description, and the uncertainty in the specifications and requirements. Precise information is usually impossible to obtain. As the design proceeds from the preliminary stage to detailed design and analysis, the level of imprecision is reduced. Finally, the design description will be precise, except for tolerances, which represent the allowable uncontrolled manufacturing variation. Despite the unavoidable imprecision in the preliminary design stage, engineering design methods and computer aids require precise information. The M_OI was developed to represent and manipulate the imprecise information in the preliminary design stage because the designer faces the highest imprecision, and the most expensive decisions are made, in the preliminary stage [21, 56, 60, 57].

An imprecise variable in the preliminary design may potentially take on any value within a possible range. Although the nominal value of the imprecise variable is unknown, some values are preferred more than others by the designer. The *method of imprecision* borrows the notion of membership functions in a fuzzy set to represent the preference among designs. Although the preference function in the M_OI and the membership function in the fuzzy sets both have values from 0 to 1.0, they are different. The membership function models the uncertainty in categorization. The preference function is fuzzy in unresolved alternatives.

Many engineering design tools, such as finite element models, require precise specification. They can be used to evaluate designs one by one. Optimization can be used to find the single

“optimal” design. But the information is only available near that single point. In contrast, the M_0I is a *set-based* method. Sets of designs are evaluated in the M_0I . The case study of Toyota’s design and development process shows that set-based methods enable effective communication, allow greater parallelism, and permit early decisions based on information that is not yet precise [59, 58, 32, 55].

In the M_0I , the design engineers identify preferences on each of the performance variables by which each design alternative will be evaluated. These preferences will typically come from potential customers. The designers also identify preferences on design variables (dimensions, material properties, *etc.*). These preferences will come from the designers’ experience and judgment, and are subject to change as the design process proceeds. One of the central aspects of the M_0I is mapping the preferences from design variables onto the performance variables, and then building an aggregate overall preference.

Many people have contributed to the M_0I , and one design tool (IDT) was built by William S. Law [27, 28]. In his Ph.D. thesis, William asked several questions about the implementation of the M_0I based on an example in [26]. This example is the mapping of a rectangle in the plane of two design variables x_9 and x_{10} to the plane of two performance variables bending stiffness K_B and torsional stiffness K_T . The approximation of K_B is shown in Equation 1.1, and the approximation of K_T is shown in Equation 1.2.

$$K_B = 78,400 + 170x_9 - 240x_{10} - 630x_9^2 - 5x_9x_{10} - 88x_{10}^2 \quad (1.1)$$

$$K_T = 13,300 + 130x_9 - 38x_{10} - 620x_9^2 + 5x_9x_{10} + 4x_{10}^2 \quad (1.2)$$

The plane of x_9 and x_{10} is shown in Figure 1.1. The center points, the four corner points, four center points on the boundaries, and the boundary are mapped to the plane of K_B and K_T . The results of the mapping are shown in Figure 1.2. The solid lines are the mapping of the boundary in Figure 1.1. The dashed lines connect the mapping of four corner points. The maximum of K_B is found at $(0.05 - 1.36)$ by optimization. Because of the nonlinearity of K_B and K_T , the boundary of the mapped rectangle is not only curved, but also crosses over itself. There are many ways to approximate the actual mapping: connecting the mapping of the four corner points by linear approximations of K_B and K_T ; connecting the mapping of eight points on the boundary of the rectangle of x_9 and x_{10} and the mapping of the center points can be used to indicate the nonlinearity of K_B and K_T ; or constructing a boundary box with the extrema of K_B and K_T found

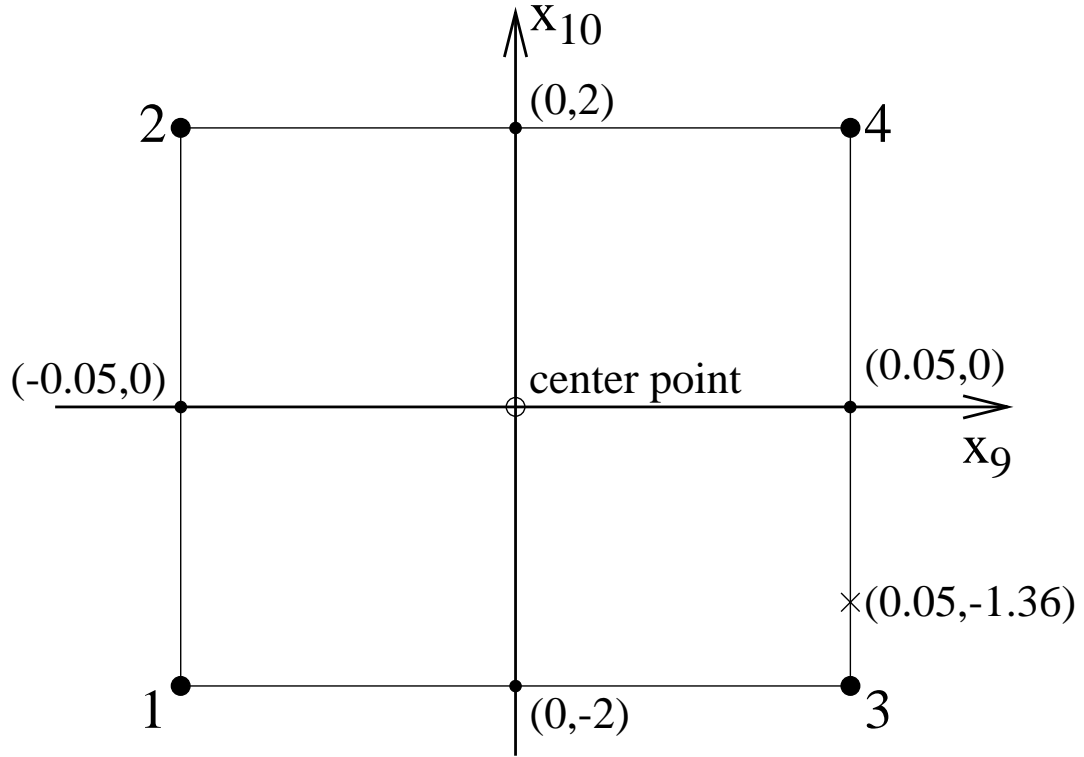


Figure 1.1: The rectangle in the plane of x_9 and x_{10} .

by optimizations. Each approximation listed above has its advantages and disadvantages.

Several questions generated from this example need to be answered:

1. Is the linear approximation sufficiently accurate for preliminary engineering design?
2. Will a nonlinear approximation of the mapping functions increase the accuracy of the boundary, but not increase the computation cost significantly?
3. Which approximations are the most accurate and the most flexible among the three approximations discussed above, or is there any other method to approximate the boundary?
4. Is there any way to let the designer make a compromise or trade-off between the cost and the accuracy with which the boundary is approximated?

1.1 Organization of Thesis

This thesis builds on the work of Wood and Antonsson [61, 62, 63, 64], Otto and Antonsson [38, 40, 43, 44, 64], Law and Antonsson [27, 28, 29, 30, 31], and Scott and Antonsson [50, 49, 51, 48, 52].

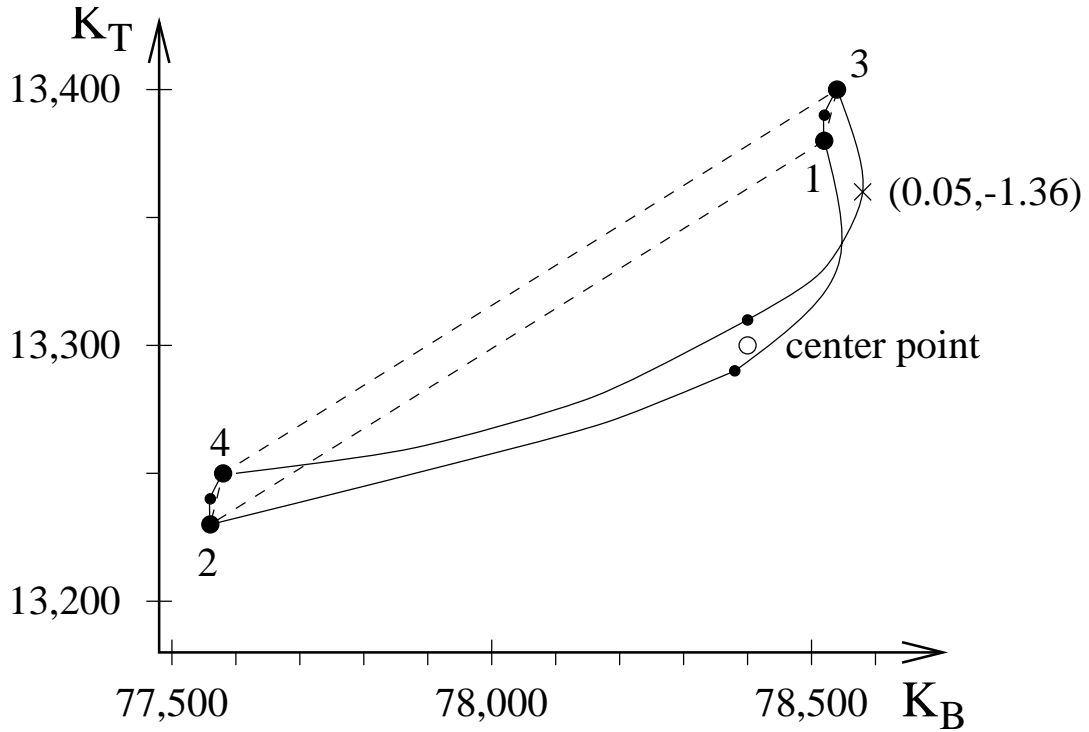


Figure 1.2: The mapping of the rectangle in the plane of K_B and K_T .

Their work has laid a broad theoretical foundation and practical implementation for the method of imprecision.

The work described in this thesis seeks to improve the accuracy and efficiency of the implementation of the M_{OI} , by practical testing via specific examples. Its principal contributions are the introduction of a multistage nonlinear metamodel into the M_{OI} , a new extension of the LIA, and a new method to compute the overall preference with loose constraints.

Chapter 2 introduces basic concepts and techniques in the M_{OI} . Section 2.1 defines the basic concepts such as variables, spaces and preferences. The overall preference, which is used to evaluate the design, is introduced in Section 2.2. Aggregation functions and rational aggregation are also discussed in the same section. Section 2.4 presents the family of rational aggregation functions.

Chapter 3 describes the process to construct multistage nonlinear metamodels. Section 3.1 discusses why the interpolation model is preferred. Section 3.2 focuses on the details of the interpolation model structure. Section 3.3 discusses the experimental designs used to choose design points for the first metamodel. Sections 3.4 and 3.5 choose the base functions in the metamodel. The sampling criterion is discussed in Section 3.6. Two methods to test the improvement of metamodels are introduced in Section 3.7. Several sampling criteria are tested on metamodels of two functions

in Sections 3.8 and 3.9.

Chapter 4 presents the efficient and accurate computation of the overall preferences. The extension principle and the LIA are introduced in Section 4.1. Then some anomalies and limitations of the original LIA implementation are discussed in Section 4.2. Section 4.3 introduces some extensions of the original LIA. The methods to compute overall preferences in both DVS and PVS without \vec{f}^{-1} are discussed in Section 4.4.

Chapter 5 presents the full new implementation of the M_OI . The models and methods discussed in Chapters 3 and 4 are combined into the new implementation of the M_OI in Section 5.1. A measure of the sensitivity of the α -cuts to the metamodel is also proposed in Section 5.1. In Sections 5.2 to 5.4, the new implementation of the M_OI is demonstrated on a practical design problem, and the results are compared with previous results.

Chapter 6 summarizes the contributions in this thesis and answers the questions asked at the beginning.

Chapter 2

The Basics of the Method of Imprecision

This chapter will focus on the basic concepts and techniques in the Method of Imprecision [62, 28]. Section 2.1 defines the basic concepts such as variables, spaces and preferences. The overall preference which is used to evaluate each design is introduced in Section 2.2. Aggregation functions and axioms for rational aggregation are also discussed in the same section. Section 2.4 presents a family of the rational aggregation functions.

2.1 The Basic Concepts

The *design variables*, $\{d_1, \dots, d_n\}$, are independent variables which differentiate alternative designs. There may be other attributes of the design which are not included in the design variables because they are not required to identify different designs. The design variables can be discrete or continuous, but they are at least ordinal in order to facilitate computations. The independence between design variables does not imply that design variables can not be related, but means that the value of each design variable can be freely chosen.

All alternative designs under consideration form the *design variable space* or *DVS*. The set of valid values for the design variable d_i is denoted \mathcal{X}_i . All design variables form an n -vector, \vec{d} , which distinguishes one particular alternative design from others in the DVS.

$$DVS = x_1 \times x_2 \times \dots \times x_n \quad [\text{Cartesian Set Product}] \quad (2.1)$$

The *performance variables*, $\{p_1, \dots, p_q\}$, are the independent variables used to indicate the performance achieved by all designs under consideration. Each performance variable is a function of the \vec{d} , $p_j = f_j(\vec{d})$. The set of valid values for a performance variable p_j is denoted \mathcal{Y}_j . All

performance variables for each alternative design form a q -vector, $\vec{p} = \vec{f}(\vec{d})$, which specifies the quantified performances of a design \vec{d} . The *performance variable space*, or *PVS*, is the set of all quantified performances achievable by all designs in DVS, where $\vec{f}(\vec{d}) = \{f_1(\vec{d}), \dots, f_q(\vec{d})\}$. The mapping function $f_j(\vec{d})$ can be any calculation, such as closed-form functions, empirical “black-box” functions, physical experiments, or even from consumer surveys.

$$PVS = \{ \vec{p} \mid \vec{p} = \vec{f}(\vec{d}), \forall \vec{d} \in DVS \} \subseteq p_1 \times p_2 \times \dots \times p_q \quad (2.2)$$

The design variables and performance variables are imprecise in nature. The final value of each variable is unspecified, and only the range of each variable is known in the preliminary stage of the design. But certain values in the range are preferred more than others. The preference can be used to quantify the imprecision of each variable.

The *functional requirement* $\mu_{p_j}(p_j)$ represents the customer’s direct preference for values of the performance variable p_j , which may be specified by customers, or estimated by the designers:

$$\mu_{p_j}(p_j) : \mathcal{Y}_j \rightarrow [0, 1] \quad (2.3)$$

The functional requirements preferences are based on quantified aspects of design performances represented by performance variables. Other unquantified aspects of design performance such as style are usually not modeled by performance variables; the preferences of these aspects are represented by the design preferences. The *design preference* function $\mu_{d_i}(d_i)$ represents the designer’s preference for values of the design variable d_i , which will be specified by the designers based on design considerations:

$$\mu_{d_i}(d_i) : \mathcal{X}_i \rightarrow [0, 1] \quad (2.4)$$

2.2 The Aggregation Functions

One single scalar preference is needed to compare different designs in DVS, where $\mu(\vec{d}_1) > \mu(\vec{d}_2)$ means that design \vec{d}_1 is more preferred than \vec{d}_2 . This single preference should embody both the design preference in the DVS and the functional requirements in the PVS. This combined preference

is called the *overall preference*, which can be expressed in the DVS as $\mu_o(\vec{d})$:

$$\mu_o(\vec{d}) = \mathcal{P}(\mu_{d_1}(d_1), \dots, \mu_{d_n}(d_n), \mu_{p_1}(f_1(\vec{d})), \dots, \mu_{p_q}(f_q(\vec{d}))) \quad (2.5)$$

The overall preference of the achievable performance can also be expressed in the PVS as $\mu_o(\vec{p})$:

$$\mu_o(\vec{d}) = \mathcal{P}(\mu_{d_1}(\vec{p}), \dots, \mu_{d_n}(\vec{p}), \mu_{p_1}(p_1), \dots, \mu_{p_q}(p_q)) \quad (2.6)$$

The \mathcal{P} in Equation 2.5 is the *aggregation function*, which reflects how the competing attributes of the design should be traded off against each other [40, 41], and formalizes the designer's balancing of conflicting goals and constraints. In order to model the designer's trade-off strategy, some restrictions must be applied on the aggregation functions to maintain their rationality [38]. These restrictions are described by the following five axioms, where $N = n + q$.

Axiom 2.1 *Commutativity*:

$$\mathcal{P}(\mu_1, \dots, \mu_j, \dots, \mu_k, \dots, \mu_N) = \mathcal{P}(\mu_1, \dots, \mu_k, \dots, \mu_j, \dots, \mu_N) \quad \forall 1 \leq j, k \leq N$$

This axiom indicates the aggregation function's independence on the order in which the individual preferences are combined.

Axiom 2.2 *Monotonicity*:

$$\mathcal{P}(\mu_1, \dots, \mu_k, \dots, \mu_N) \leq \mathcal{P}(\mu_1, \dots, \mu'_k, \dots, \mu_N) \quad \text{for } \mu_k \leq \mu'_k, \forall 1 \leq k \leq N.$$

The monotonicity means that the change of the overall preference caused by any change in any individual preference should not move in the opposite direction. If the monotonicity is not satisfied, an increase in one individual preference will cause decrease in the overall preference, which is not rational.

Axiom 2.3 *Continuity*:

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

The overall preference should not have any discontinuities if there are no discontinuities in individual preference.

Axiom 2.4 *Idempotency:*

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

The idempotency will remove the artificial biasness in the aggregation function.

Axiom 2.5 *Annihilation:*

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

A preference of zero indicates that the value of that variable is totally unacceptable. This axiom is needed to make sure that any acceptable design does not have any unacceptable design performance.

These five axioms are only necessary conditions for the rationality of the aggregation function. Similar axioms are defined by Fung and Fu [20] to maintain the rationality in general decisionmaking: commutativity, monotonicity, continuity, idempotency, and associativity. It is noted that the annihilation axiom is not necessary for the rationality of general decision-making.

The aggregation functions satisfying all above axioms are considered *design-appropriate*. Although there are many design-appropriate aggregation functions, the choice should be made according to the relationship between design and performance variables as follows [42, 64].

In one type of design strategy, the overall preference of the system is determined by the lowest preference on any variable. The increase in preference for one variable can not compensate for the decrease in another preference. There is no trade-off between individual preferences. This is a *non-compensating* trade-off strategy. The minimum aggregation function \mathcal{P}_{min} should be used here:

$$\mu_o = \min(\mu_{d_1}, \dots, \mu_{d_n}, \mu_{p_1}, \dots, \mu_{p_q}) \quad (2.7)$$

\mathcal{P}_{min} is the hard “and” operation in fuzzy logic [4], which will lead to the classic max-min solution from game theory [65].

In another type of design strategy, the decrease in one preference can be counterbalanced by the increase in another, that is, the two preferences can be traded off with each other. This is a *fully compensating* trade-off strategy. The geometric weighted mean function \mathcal{P}_{Π} is needed:

$$\mu_o = \left(\prod_{i=1}^n \mu_{d_i} \cdot \prod_{j=1}^q \mu_{p_j} \right)^{\frac{1}{n+q}} \quad (2.8)$$

\mathcal{P}_{Π} is the soft “and” operation in the fuzzy logic [4], which corresponds to the Nash solution from game theory [65].

2.3 Weights of Preferences

The variables representing a design are not equally important with each other. The relative importance of each variable can be specified by assigning *weights* to the corresponding variables:

$$\omega_{d_i} \geq 0$$

$$\omega_{p_j} \geq 0$$

Now the overall preference should be aggregated from both the individual preferences and the individual weights. The axioms which the aggregation functions should satisfy need to be redefined in order to include weights:

Axiom 2.6 *Commutativity:*

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

Axiom 2.7 *Monotonicity:*

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

The overall preference should not decrease if the weight of the variable with the highest preference is increased.

Axiom 2.8 *Continuity:*

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

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

The aggregation function should have continuities on both preferences and weights.

Axiom 2.9 *Idempotency:*

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

Axiom 2.10 *Annihilation:*

$$\mathcal{P}(\mu_1, \dots, 0, \dots, \mu_N; \omega_1, \dots, \omega_k, \dots, \omega_N) = 0 \quad \text{where } \omega_k \neq 0, \quad \forall k$$

$$\mathcal{P}(\mu_1, \dots, \mu_k, \dots, \mu_N; \omega_1, \dots, 0, \dots, \omega_N) =$$

$$\mathcal{P}(\mu_1, \dots, \mu_{k-1}, \mu_{k+1}, \dots, \mu_N; \omega_1, \dots, \omega_{k-1}, \omega_{k+1}, \dots, \omega_N) \quad \forall k$$

If a variable is assigned a weight of zero, that variable will be removed from the aggregation.

Because no upper boundaries and normalizations are specified on the weights in the definition, an additional axiom is needed:

Axiom 2.11 *Self-normalization:*

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

According to the self-normalization axiom, the weights can be scaled by any positive constant λ without any change in the overall preference. A weighted aggregation function is design appropriate if all above six axioms are satisfied.

2.4 Weighted Means

The theory of functional equations [1] has been applied to the exploration of certain aggregation functions called t-norms and t-conorms [14]. The same approach was applied to the design-appropriate aggregation functions by Scott and Antonsson [50]. The relevant class of functions is the weighted means:

$$\mathcal{P}(\mu_1, \dots, \mu_N; \omega_1, \dots, \omega_N) = g \left(\frac{\sum_{i=1}^N \omega_i g^{-1}(\mu_i)}{\sum_{i=1}^N \omega_i} \right) \quad (2.9)$$

where g is a strictly monotonic, continuous function with inverse g^{-1} ; $g(0) \leq \mu_i \leq g(1)$, $\omega_i \geq 0$, $1 \leq i \leq N$; and $\sum_{i=1}^N \omega_i > 0$. It is shown that the weighted means satisfy all the axioms except possibly the annihilation axiom. Then all weighted means which satisfy the annihilation axiom are design-appropriate. Arbitrary design-appropriate aggregation functions also satisfy other properties with the assumption of strict monotonicity [50]. Any strictly monotonic design-appropriate aggregation must be a weighted mean.

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

$$\mathcal{P}_s(\mu_1, \dots, \mu_N; \omega_1, \dots, \omega_N) = \left(\frac{\sum_{i=1}^N \omega_i \mu_i^s}{\sum_{i=1}^N \omega_i} \right)^{\frac{1}{s}} \quad (2.10)$$

where $s \in \mathcal{R}$. If $s > 0$, $0 = g(0) \leq \mu_i \leq g(1) = 1$. If $s < 0$, $g^{-1}(\mu)$ and \mathcal{P}_s satisfies the annihilation axiom [50]. $\mathcal{P}_{s \leq 0}$ is a family of design-appropriate aggregation functions because it satisfies all the axioms in Section 2.3.

The two design-appropriate aggregation functions, \mathcal{P}_{min} and \mathcal{P}_{Π} , are just limiting cases of $\mathcal{P}_{s \leq 0}$ where $s \rightarrow 0$ & $s \rightarrow -\infty$ respectively [50]:

$$\mathcal{P}_{\Pi} = \mathcal{P}_{s=0}(\mu_1, \dots, \mu_N; \omega_1, \dots, \omega_N) = \left(\prod_{i=1}^N \mu_{d_i}^{\omega_i} \right)^{\frac{1}{\omega}} \quad \text{where } \omega = \sum_{i=1}^N \omega_i \quad (2.11)$$

and

$$\mathcal{P}_{s=-\infty}(\mu_1, \dots, \mu_N; \omega_1, \dots, \omega_N) = \min(\mu_1, \dots, \mu_N) \quad (2.12)$$

Other \mathcal{P}_s with $-\infty < s \leq 0$ changes continuously with s between \mathcal{P}_{min} and \mathcal{P}_{Π} and represent *partially compensating* trade-off strategies, where a change of preference for one variable can be partially compensated for by changing the preference for another variable.

Because of the properties of idempotency and monotonicity, \mathcal{P}_{min} is the lower bound for design-appropriate functions. Similarly, $\mathcal{P}_{max} = \max(\mu_1, \dots, \mu_N)$ might be the upper bound for design-appropriate functions. But \mathcal{P}_{max} is not design-appropriate because it does not satisfy the annihilation axiom.

For \mathcal{P}_s with $s \geq 0$, the level of compensation increases with s until $\mathcal{P}_{s=+\infty} = \max$ is reached. $\mathcal{P}_{s>0}$ are also called *super-compensating* functions. Just like \mathcal{P}_{max} , all super-compensating functions do not satisfy the annihilation axiom. But they can be modified to become design-appropriate

because of the actual implementation of the M_OI [48].

The parameter s defines the trade-off strategy or degree of compensation between any two variables, and is implemented by a design-appropriate aggregation function $\mathcal{P}_s(\mu_1, \mu_2; \omega_1, \omega_2)$. The *indifference points* are defined as two points which have the same preference. The parameter s and the weights can be numerically calculated from indifference points [52].

2.5 Hierarchical Aggregation

If many different trade-off strategies are used to aggregate the overall preference in a design, the individual preferences should be aggregated by an aggregation hierarchy. The hierarchy is determined by the problem. Even if different \mathcal{P}_s 's are used in the hierarchy, the weights can be propagated freely because of the self-normalization axiom discussed in Section 2.3. If there are only aggregations of pairs of individual preferences in the hierarchy, a numerical method can be used to calculate the parameter s and the weights [52].

One special situation for hierarchical aggregation is aggregations with the same s that are combined together. Consider three individual preferences μ_1 , μ_2 and μ_3 with weights ω_1 , ω_2 and ω_3 . Assume μ_1 is first aggregated with μ_2 , then aggregated with μ_3 with the same trade-off strategy.

$$\begin{aligned}
 & \mathcal{P}_s(\mathcal{P}_s(\mu_1, \mu_2; \omega_1, \omega_2), \mu_3; \omega_1 + \omega_2, \omega_3) \\
 &= \left(\frac{(\omega_1 + \omega_2) \mathcal{P}_s(\mu_1, \mu_2; \omega_1, \omega_2)^s + \omega_3 \mu_3^s}{(\omega_1 + \omega_2) + \omega_3} \right)^{\frac{1}{s}} \\
 &= \left(\frac{(\omega_1 + \omega_2) \left(\frac{(\omega_1 \mu_1^s + \omega_2 \mu_2^s)^{\frac{1}{s}}}{\omega_1 + \omega_2} \right)^s + \omega_3 \mu_3^s}{\omega_1 + \omega_2 + \omega_3} \right)^{\frac{1}{s}} \\
 &= \left(\frac{\omega_1 \mu_1^s + \omega_2 \mu_2^s + \omega_3 \mu_3^s}{\omega_1 + \omega_2 + \omega_3} \right)^{\frac{1}{s}} \\
 &= \mathcal{P}_s(\mu_1, \mu_2, \mu_3; \omega_1, \omega_2, \omega_3)
 \end{aligned} \tag{2.13}$$

Therefore in this case, the hierarchical aggregation is unnecessary since it is equivalent to the single aggregation function.

2.6 Summary

All designs under consideration form the DVS, and the design variables, are used to distinguish alternative designs. The PVS consists of all achievable performances. The performance variables can be mapped from the vector of design variables. All quantified preferences are specified on the performance variables directly, usually in consultation, or by survey, of the customers. Other unquantified preferences are determined by the designer based on judgment and experience.

The design will be evaluated by the overall preference which is aggregated from individual preferences. A design with higher overall preference is preferred more than one with lower overall preference. The aggregation functions embody the trade-off strategies. All design-appropriate aggregation functions represent rational trade-off strategies and satisfy the five axioms: commutativity, monotonicity, continuity, annihilation, and idempotency, among which the annihilation is unique to design problems.

Weights may be included in the aggregation function in order to represent the relative importance of individual preferences. In this case, all of the above five axioms need to be redefined, and one self-normalization axiom is added for the scalability of weights.

The family of root-mean-power functions \mathcal{P}_s with a negative parameters, $\mathcal{P}_{-\infty < s \leq 0}$, contain all monotonic design-appropriate aggregation functions. The root-mean-power function \mathcal{P}_s with positive parameter $s > 0$ is not design appropriate, but it can be modified to be design appropriate because of the implementation details of the M_OI, as shown later. If different trade-off strategies are used in a design, a hierarchy of aggregation is needed.

This completes the introduction to the basic model structure and techniques used in the M_OI. The computational implementation of M_OI will be discussed in later chapters.

Chapter 3

Metamodels for the Mapping between DVS and PVS

A central element of the M₀I, is to approximate the mapping between the design variable space (DVS) and the performance variable space (PVS), because the mapping will be used to induce preferences in the PVS from the preferences in the DVS, as will be discussed in Chapter 4. Because of the wide use of computers, this process is usually conducted by running some complex computer analysis software package. When the DVS is high dimensional, it is prohibitively expensive to directly use complex analysis software to explore the DVS. For example, it will require about 5 minutes to analyze a modestly complex finite element model. For a five dimensional DVS, it will require about 10 days on a grid with 5 points on each design variable (DV). If the dimensionality of the DVS increases to 10, then it will run about 92 years, on a grid with 5 points on each DV.

It is not necessary to pursue high accuracy at the preliminary stage of the engineering design. So, a linear approximation can often be used to reduce the computational cost [30]. It does not perform well on nonlinear mappings common in design, so a traditional optimization is used to improve the accuracy [30]. It is preferred to have one single model to estimate linear and nonlinear mappings.

A *metamodel* is defined as “a model of the model [24].” It should have flexible model structure and be able to estimate the actual mapping with reasonable accuracy. A metamodel will be constructed by running the analysis software over a relatively small set of design points, and will be used to explore the mapping between the DVS and the PVS. Due to the iterative nature of engineering design, multistage metamodels are helpful, because a more accurate metamodel may be needed as the design is refined.

This chapter will introduce the model structure of the metamodel in Sections 3.1 and 3.2. Section 3.3 will discuss the experimental design used to determine the design points for the metamodel

at the first stage. In Sections 3.4 and 3.5, the base functions used in the metamodel will be determined by tests on two specific examples. The sampling criteria of the design points, when there is *a priori* knowledge available, will be discussed in Section 3.6. Section 3.7 introduces two methods to test the improvements between metamodel at different stages. Finally, the sampling criteria will be tested on two specific examples in Sections 3.8 and 3.9.

3.1 Approximation or Interpolation

The difference between a computer experiment and a traditional physical experiment is that repeated computer experiments generate the same results.

For an approximate model of a traditional physical experiment, the most frequently used method is least-squares regression, which models the random errors in the results as identical and independent Normal variables with mean zero and variance σ , that is, $\varepsilon_{random} \sim N(0, \sigma^2)$. The least-square estimator will minimize the sum of the squared differences between the experiment results and the predicted values. Figure 3.1 shows one set of sample data and the results of two different approximate models. The relationship between the estimate and the actual value is

$$y = \hat{y} + \varepsilon_{system} + \varepsilon_{random} \quad (3.1)$$

where y is the actual value, \hat{y} is the estimated value, ε_{system} is the systematic error, and ε_{random} is the random error, or call it approximation error.

Although the estimated model does not pass through the actual values, it is assumed that the random errors are smoothed out because of the assumption of $\varepsilon_{random} \sim N(0, \sigma^2)$

However, for the deterministic results from a computer experiment, the relationship between the estimate and the actual value is

$$y = \hat{y} + \varepsilon_{system} \quad (3.2)$$

where y is the actual value, \hat{y} is the estimated value, and ε_{system} is the systematic error.

Although the response surface method is based on least square regression, it still can be applied to computer experiments if the dimensionality of the design space is not high and the response is not strongly nonlinear, because of the simplicity and maturity of the response surface method.

Neural networks implement a nonlinear regression method [8]. It may be used for deterministic

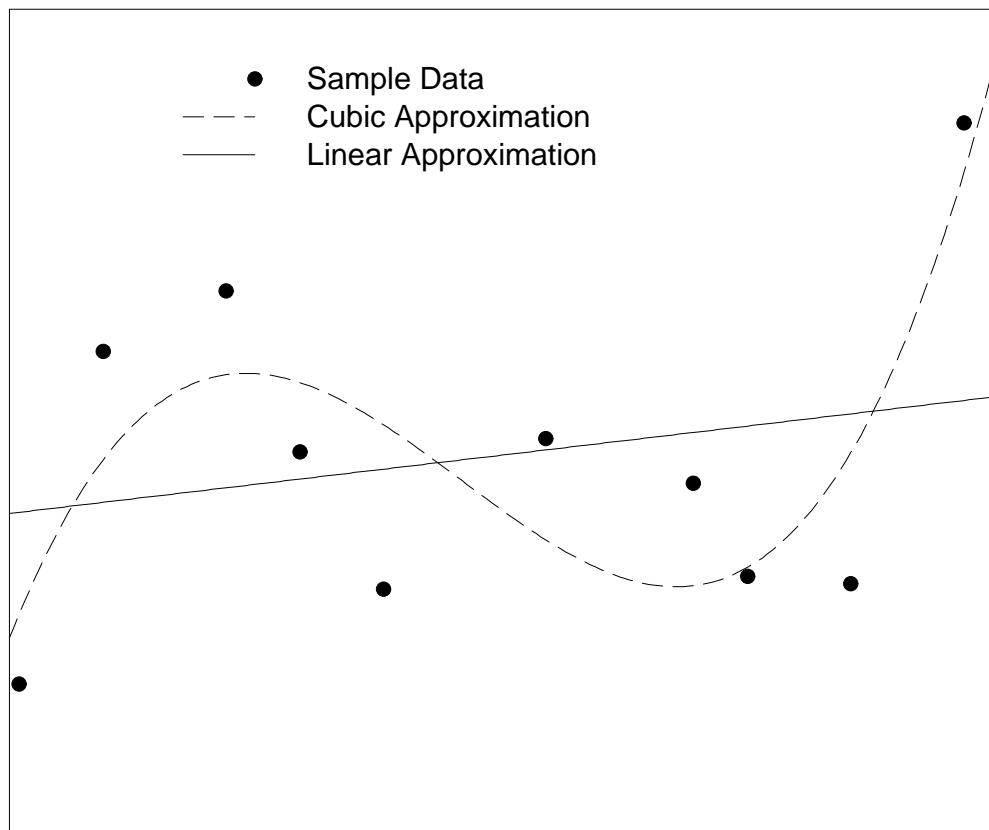


Figure 3.1: Two approximations of one set of sample data.

high-dimensional computer experiments. But the fatal disadvantage of the neural networks is that it converges slowly. To construct a good metamodel, it needs lots of training data. This makes the neural networks not suitable for building the metamodel in the preliminary stage of engineering design where available resources are limited.

Inductive learning draws inductive inference from obtained facts [17, 16]. This method builds the estimation model in the form of condition-action rules, and decision trees. A matched rule is found by a search in the decision tree for the encountered condition [25]. This method works well if the design variables and the response are almost all discrete-valued, but it is not a good candidate in engineering design where, in general, the responses are continuous

By probabilistic modeling of the uncertain prediction error, J. Sacks *et al.* proposed a new interpolation model which generates the best unbiased linear predictor for the deterministic computer experiment [46]. This method is discussed in detail in the next section.

3.2 Interpolation Model

Consider the approximation of $Y(x)$ by some $\hat{Y}(x)$ where $x \in \mathcal{R}^m$. In Equation 3.2, the structure of the systematic error ϵ_{system} is usually unknown. In the approach proposed by Sacks, Schiller and Welch [46], the estimate is an approximation, $\sum_{j=1}^k \beta_j \cdot f_j(x)$, and ϵ_{system} is modeled as a stochastic process, $Z(x)$. Now the model becomes

$$Y(x) = \sum_{j=1}^k \beta_j \cdot f_j(x) + Z(x) \quad (3.3)$$

The approximation is simple and straightforward once the choice of the base functions f_j is made. For the stochastic process $Z(x)$, the important part is the covariance structure, and this is chosen to be

$$\begin{aligned} Cov(Z(t), Z(u)) &= V(t, u) \\ &= \sigma_z^2 \cdot \exp(-\theta \cdot \sum_{j=1}^m (t_j - u_j)^{p_j}) \end{aligned} \quad (3.4)$$

where $t = (t_1, \dots, t_m)$ and $u = (u_1, \dots, u_m)$, σ_z^2 , p_j and $\theta > 0$ are parameters to be decided by the designer.

It is always assumed that $E(Z(x)) = 0$ and that $Z(x)$ is a Gaussian process. The value of θ will

affect the prediction ability of the model. It is harder to predict for the model with larger θ than the model with smaller θ . The choices of p_j 's will determine the derivatives of the correlation functions and the response. Here the p_j 's are chosen to be 2, and θ is set to $1/2$.

The points in the DVS used to build the estimation model are called the *design points*. With a set of n design points, $S = \{s_1, \dots, s_n\}$ and corresponding response $Y(s_1), \dots, Y(s_n)$, the interpolation model is generated as follows by Sacks, Schiller and Welch [46]. First introduce the notation:

$$\begin{aligned}
 \beta &= [\beta_1, \dots, \beta_k] \\
 V &= [Cov(Y(s_i), Y(s_j))]_{1 \leq i \leq n, 1 \leq j \leq n} \\
 f'_x &= [f_1(x), \dots, f_k(x)] \\
 v'_x &= [V(s_1, x), \dots, V(s_n, x)] \\
 y' &= [Y(s_1), \dots, Y(s_n)] \\
 F &= [f_l(s_i)]_{1 \leq i \leq n, 1 \leq l \leq k}
 \end{aligned} \tag{3.5}$$

For the linear predictor of the response, $c' \cdot y$, its mean square error (MSE) is [46]:

$$E[c' \cdot y - Y(x)]^2 = (c' \cdot F \cdot \beta - f'_x \cdot \beta)^2 + [c', -1] \cdot \begin{bmatrix} V & v_x \\ v'_x & \sigma_z^2 \end{bmatrix} \cdot \begin{bmatrix} c \\ -1 \end{bmatrix} \tag{3.6}$$

To obtain an unbiased predictor, it is needed to apply the constraint $F' \cdot c = f_x$. After minimizing the MSE for the predictor $c' \cdot y$ under the above constraint by Lagrange multipliers method, the interpolation model becomes [46]

$$\begin{aligned}
 \hat{Y}(x) &= f'_x \cdot \hat{\beta} + v'_x \cdot V^{-1} \cdot (y - F \cdot \hat{\beta}) \\
 \hat{\beta} &= (F' \cdot V^{-1} \cdot F)^{-1} \cdot F' \cdot V^{-1} \cdot y
 \end{aligned} \tag{3.7}$$

where $\hat{\beta}$ is the generalized least square estimate of β .

3.3 Experimental Design

Experimental design methods have been widely used in many areas including computer experiments. It helps to learn how systems work. Careful design of experiments will result in improved process yields, reduced development costs, etc. It also has an important role in the area of engineering design. It helps the designer to compare different design configurations, to optimize design parameters, to improve robustness of the design, etc. The design of experiment also helps to choose design points which are used to construct the metamodel efficiently.

There are many frequently used experimental design methods for different purposes. One such method is Latin Hypercube sampling first introduced by M.D. McKay *et. al.* [35], which ensures all portions of the design space are sampled. The Latin Hypercube method is an extension of stratified sampling. With stratified sampling, the design space is divided into many disjoint strata. Each stratum is sampled individually. The Latin Hypercube simply divides the region of each design variable into N strata with equal marginal probability distribution. If the design variables are distributed equally, each stratum will be divided into the same range, and then the method is called Uniform Latin Hypercube sampling. Each stratum is sampled only once. Therefore, for the i^{th} design variable, there will be N different sampling locations $\{dv_{i,1}, \dots, dv_{i,N}\}$. The values of the i^{th} design variable at all N sampling points, $\{x_{i,1}, \dots, x_{i,N}\}$, will be a permutation of $\{dv_{i,1}, \dots, dv_{i,N}\}$. Because all portions of each design variable region are covered in the Latin Hypercube sampling, it is best used when there are only a few dominating design variables for the response. The Latin Hypercube method usually needs a reasonably large number of samples to make the method work well.

In a *factorial design*, all possible combination of the levels of each factor are evaluated [37, Page 228]. In general, factorial design is the most efficient method if it is desired to study the effects of more than one factor. Factorial design allows not only study of the effects of the main factor, but also the effects of the interactions between factors. The most important factorial design method is the 2-level factorial design, which requires 2^k observations for k factors. It is also called the 2^k *factorial design* [37, Page 290]. The 2^k factorial design works quite well in the studies of the linear effects of the main factors and the interactions. However, the number of required observations increases dramatically when the number of factors increases. If it can be assumed that certain high-order interactions are negligible, then only a fraction of the 2^k factorial design is needed to get the information of the main factors and the low-order interactions. The high-order interactions with

fixed values are called the generators. With P independent generators, the 2^{k-p} *fractional factorial design* can be constructed, which is $1/2^p$ fraction of the 2^k factorial design [37, Page 398]. There are many possible choices of generators, but only some of them can generate the highest possible resolutions. If two or more effects can not be differentiated by the observations, they are called *aliases* [37, Page 374]. The resolution of a fractional factorial design is considered to be K if n -factor effects and effects with less than $(K - n)$ -factor are not aliases [37, Page 376]. The most useful ones are Resolution III designs, Resolution IV designs, and Resolution V designs.

For a Resolution III design, no two or more main effects are aliases, but any main effect and any two-way interactions may be aliases. For a Resolution IV design, no two or more main effects are aliases, neither are any main effects and any two-way interaction. But two or more two-way interactions may be aliases. For a Resolution V design, no two or more main effects are aliases, neither are two or more two-way interactions. But any two-way interaction and any three-way interaction may be aliases [9].

If the regression model is only first-order, the *orthogonal first-order designs* [37, Page 600] can be used to minimize the variance of the regression parameters. A design is orthogonal if the matrix $(X'X)$ is diagonal. The 2^k factorial design and fractional factorial design are both orthogonal. Another type of orthogonal design is the *simplex design* [11]. The simplex design is a equilateral triangle for $k = 2$, and is a regular tetrahedron for $k = 3$.

For fitting the second-order polynomial regression model, the *central composite design*, or *CCD* [37, Page 601], is the most popular design. Usually the *CCD* consists of a Resolution V fractional factorial design, the center point, and $2 \cdot k$ axial points for k factors. There are several variations of *CCD*. If the fractional factorial design in the *CCD* is only Resolution III, the design is called *small composite design* [37, Page 605]. Sometimes the interesting region is the k -dimensional hypercube, then the axial points can be put at the center of each face, *i.e.*, ± 1 . This design is called the *face-centered central composite design* [37, Page 605].

The Box-Behnken design, proposed by Box and Behnken, is also used for the second-order regression model [5]. It is constructed by combining the 2^k factorial design and the incomplete block design. This type of design is spherical design. All the design points are on a sphere of radius $\sqrt{2}$, and there are no corner points included.

Among all these design methods, the right choice should be made based on the regression model and any specific requirement. For the computer experiment at the preliminary stage of engineering design, more weight will be put on parsimony, *i.e.*, fewer design points for the same number of

factors.

3.4 Base Functions

For the first part of Equation 3.3, the base functions $f_j(x)$ are unspecified. The second part of the Equation 3.3 models the error in the approximation of the response. The general principles of choosing a good model type are flexibility and parsimony, and a trade-off is always made between them. In the preliminary stage of engineering design, the computational cost is a significant concern, so more weight is put on the principle of parsimony when considering the model type of base functions. Also, there is the “main effects principle,” which is the empirical observation that linear main factors are more important than high-order interactions [34]. From all of the above considerations, several polynomial models (from the simple linear model, quadratic model, to more complicated nonlinear MARS model) are candidates for the base functions in the interpolation model.

3.4.1 Polynomial Models

The linear model of n independent variables with up to m order interactions is

$$\hat{y}_{linear} = \hat{f}(x_1, \dots, x_n) = a_0 + \sum_{i=1}^n a_i x_i \quad (3.8)$$

The design points for the linear model are determined by a Resolution III fractional factorial design.

In piecewise linear model, the design variable space is divided into many rectangular subspaces and each subspace has its own linear model, as above.

The (partially) quadratic model of n independent variables with up to m order interactions adds quadratic terms to the linear model above:

$$\hat{y}_{quad} = \hat{f}(x_1, \dots, x_n) = \hat{y}_{linear}(x_1, \dots, x_n) + \sum_{i=1}^n a_{ii} x_i^2 \quad (3.9)$$

The central composite design is used to decide the coefficients of the quadratic terms. It is simply the factorial design plus the central points of each face.

The higher-order model of n independent variables, with higher-order interactions determined

by the values of n and m , is

$$\begin{aligned} \hat{y} = \hat{f}(x_1, \dots, x_n) = & a_0 + \sum_{i=1}^n a_i x_i + \sum_{i_1=1}^n \sum_{i_2=1}^n a_{i_1 i_2} x_{i_1} x_{i_2} + \dots, \\ & + \sum_{i_1=1}^n \dots \sum_{i_m=1}^n a_{i_1 \dots i_m} x_{i_1} \dots x_{i_m} \end{aligned} \quad (3.10)$$

This is equivalent to a product of several linear regression polynomials. If the number of data points is less than the number of terms in the polynomial, only the coefficients of an equal number of lower-order terms are nonzero.

3.4.2 Nonlinear Regression MARS Model

MARS [19, 45] fits high-dimensional data to an expansion in multivariate spline basis functions. The number of basis functions, the product degree, and the knot locations are automatically determined by, and are adaptive to, the data. The model produces a strictly continuous approximation with continuous derivatives, and identifies the contributions from additive terms and multivariable interactions. The method is attractive due to its low computational cost.

“The approximation takes the form of an expansion in multivariate spline basis functions:

$$\hat{y} = \hat{f}(x_1, \dots, x_n) = a_0 + \sum_{m=1}^M a_m B_m(x_1, \dots, x_n) \quad (3.11)$$

with:

$$B_0(x_1, \dots, x_n) = 1, \quad (3.12)$$

$$B_m(x_1, \dots, x_n) = \prod_{k=1}^{K_m} b_{km}(x_{v(k,m)} | t_{km}). \quad (3.13)$$

The $\{a_m\}_0^M$ are the coefficients of the expansion. Each multivariate spline basis function B_m is the product of univariate spline basis functions b , each of a single input variable $x_{v(k,m)}$, and characterized by a knot at t_{km} . The multivariate spline basis functions B_m are adaptive in that the number of factors K_m , the variable set $V(m) = \{v(k, m)\}_1^{K_m}$, and the parameter set t_{km} are all determined by the data.” [18, Page 17]

The “knots” here means a nondecreasing sequence of t_{km} ’s which determine the control (design) points of the spline. For further details of the MARS model see Friedman (1991) [19].

Based on modeling of five-dimensional input variables on an engineering workstation, results can be obtained essentially immediately when 32 observations are used and a maximum of 30 base functions are allowed. The regression computations required less than 80 seconds for modeling and prediction based on 3125 data points with a maximum of 40 base functions.

Another advantage of MARS is that it provides methods of slicing up the n -dimensional space by assigning specific values to a subset of the design variables and obtaining the MARS model along the slice. This is of great convenience when the shape of performance response in one or two specific directions is needed.

3.5 Method for Selection of the Base Functions

Among the candidates listed in Section 3.4, one type of base function will be chosen. The choice will be made based on the performance of each type of base functions on a practical problem described below. One estimation model will be built for each type of base function using the least-squares regression method. This finite element model has already been run on a 5^5 grid in the DVS. There are $5^5 = 3125$ points in the DVS and corresponding values of the response at these points. Only a fraction of the results will be used to build the estimation model, but all data will be used to evaluate the performance of the resulting model. All the points used to test the model are called the *test points*. For each estimation model, the empirical root-mean-square error (ERMSE) at all the design points and all the test points, the maximum error at all design points and test points, and the error at the point with maximum response will be computed as the measure of the performance of each type of base function. Also, the cost of building the model will be taken into consideration. Because the computation cost of the least square regression method is negligible when compared with the cost of the finite element model, only the latter cost, *i.e.*, the number of design points, will be considered.

3.5.1 Problem Description

The test function in the example presented here is the bending stiffness of a Volkswagen passenger automobile chassis (shown in Figure 3.2) computed from a finite-element model (shown in Figure 3.3) in a design space of five variables ($n = 5$):

$$x_1 = \text{A Pillar Thickness [mm]}$$

$$x_2 = \text{B Pillar Thickness [mm]}$$

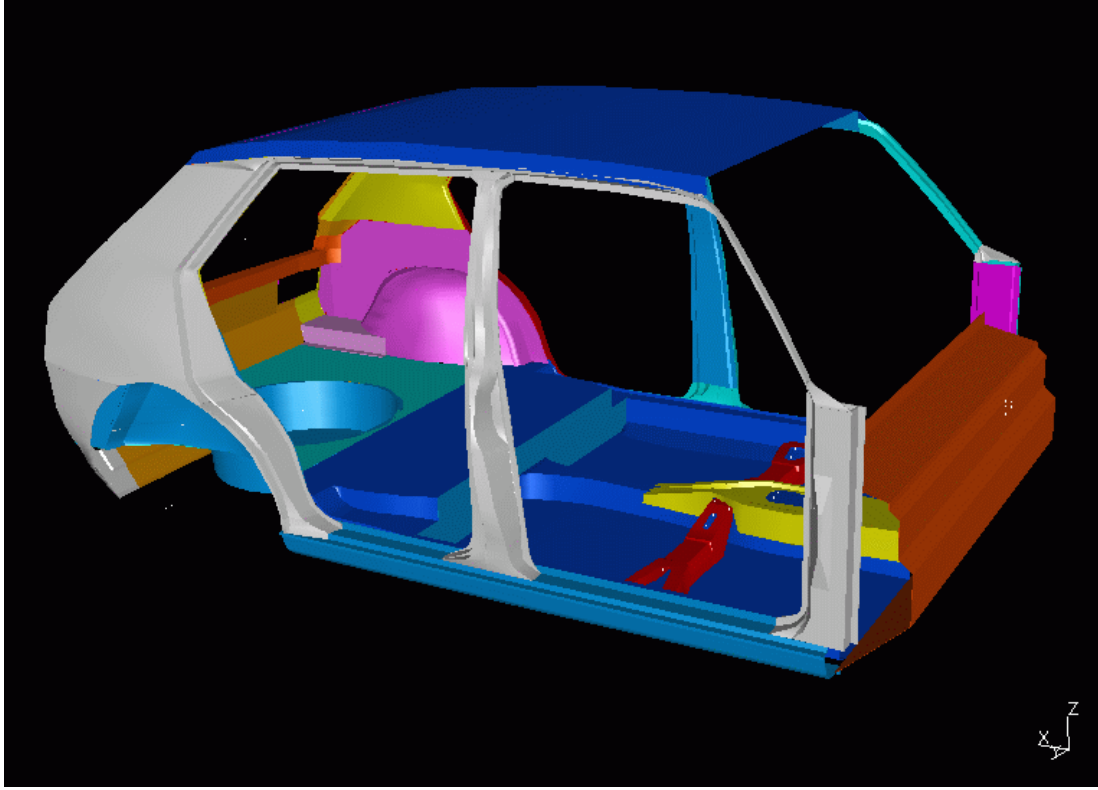


Figure 3.2: Geometric model of body-in-white in SDRC I-DEAS.

x_3 = Floor Rail Thickness [mm]

x_4 = Floor Thickness [mm]

x_5 = B Pillar Location [mm]

Table 3.1 lists the models that have been built to fit the actual response function. In all cases, $n=5$. For the linear model (LM), $m = 2$ was used for 17 points which are a Resolution IV fractional factorial design and the center point, and $m = 5$ was used for 32 points which are the full factorial design. For the quadratic model (QDM), $m = 2$ for 14 and 19 points. The 14 points are a Resolution III fractional factorial design, the center point, and five face-center points $\{d_1, \dots, d_5\}$ where $d_k = 1$ and $d_l = 0$, $\forall l \neq k, 1 \leq k \leq 5$. The 19 points are a Resolution III fractional factorial design, the center point, and ten face-center points $\{d_1, \dots, d_5\}$ where $d_k = \pm 1$ and $d_l = 0$, $\forall l \neq k, 1 \leq k \leq 5$. For the high-order polynomial model (HM), $m = 4$ was used for 34 and 106 points which is the combination of fractional factorial designs in each sub-hypercube over the grid, and $m = 10$ for 243 points which is a 3-level full factorial design. For the piecewise linear model (PLM), $m = 2$ was used for each linear model. There are 32 linear models in the PLM

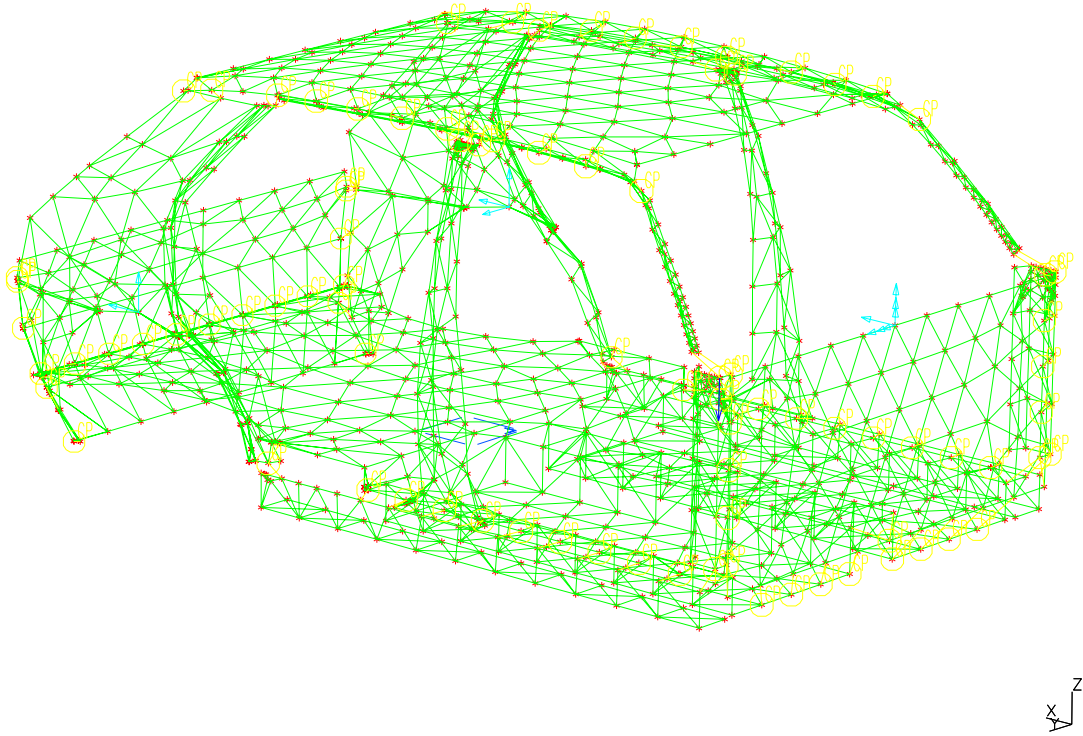


Figure 3.3: Finite element model of body-in-white.

with 243 points which is the combination of 32 2-level full factorial designs in 32 sub-hypercubes, and 1024 linear models in the PLM with 3125 points which is the combination of 1024 2-level full factorial designs in 1024 sub-hypercubes. The test points for each model are all points on the 5^5 grid except the design points.

3.5.2 Test Results

Figure 3.4 shows a projection of the five-dimensional bending stiffness data surface onto 2 dimensions (*Floor thickness* and *B pillar location*). This illustrative projection was created by holding each of the 3 dimensions not shown (*A pillar thickness*, *B pillar thickness* and *Floor rail thickness*) at a constant value. The approximations computed here fit all 5 dimensions of the design space, however, to enable graphical comparison, the figures only show variations in bending stiffness as a function of 2 dimensions.

The bending stiffness response in the *B pillar location* direction has the highest nonlinearity of the 5 directions. The nonlinearity in that direction is reflected, to some degree, in the errors of most regression models. Polynomial models and the MARS model were fitted to the data selected by each design of experiment listed in Table 3.1. Figures 3.5 through 3.7 show the error surfaces of

Table 3.1: Data Fitting Models.

Model	Experimental Design	No. of Design Points for a Single Model	No. of Models
QDM, MARS	Resolution III + half central pts	14	1
LM, MARS	Resolution V + all central pts	17	1
QDM, MARS	Resolution III	19	1
LM, MARS	Full	32	1
HM		34	1
HM		106	1
PLM	Full	243	32
HM		243	1
MARS	Full	243	1
PLM	Full	3125	1024
MARS	Full	3125	1

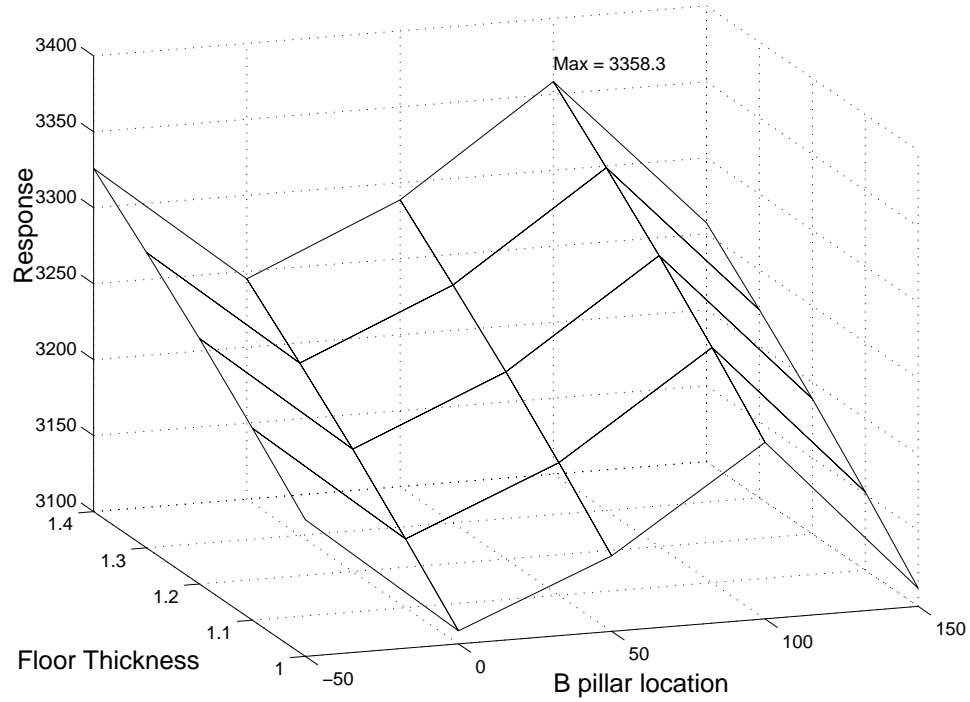


Figure 3.4: Bending stiffness data projected onto 2 dimensions.

the polynomial and MARS models with 19,243 and 3125 evaluations. The errors are computed by systematically computing the bending stiffness at 5 equally spaced points in each of the 5 variables, thus producing a five-dimensional set of 3125 data points. The error is the difference between each approximation model and the 3125 computed data points. As before, these figures are plotted by projecting the error onto two directions (*Floor thickness* and *B pillar location*) of the design space while fixing the values in each of the other three directions.

Table 3.2 shows some numerical results.

Figures 3.8 through 3.10 show error statistics from the regression models. In each case, the empirical root-mean-square error is computed as follows:

$$\text{ERMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (\text{error}_i)^2} \quad (3.14)$$

Figure 3.8 shows the mean square error computed from the difference between the computed bending stress and the regression models at the points used to build the regression model. This illustrates how close each regression model is to the known data. Figure 3.9 shows the mean square error computed at the balance of the five-dimensional set of 3125 data points not used to build the regression

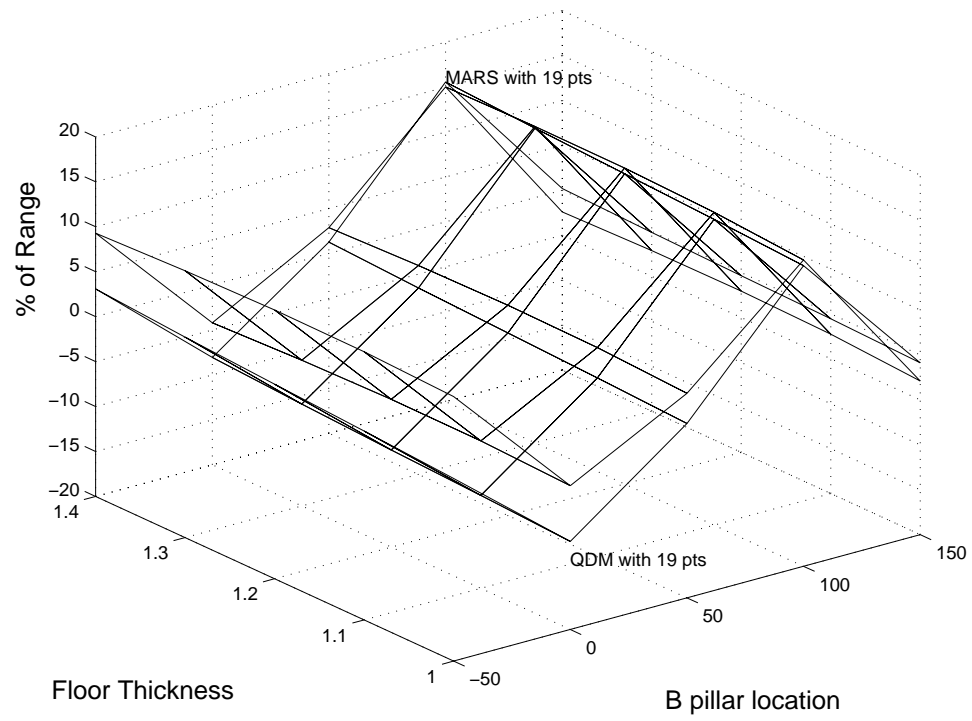


Figure 3.5: (ERMSE/response range) of polynomial and MARS models with 19 design points.

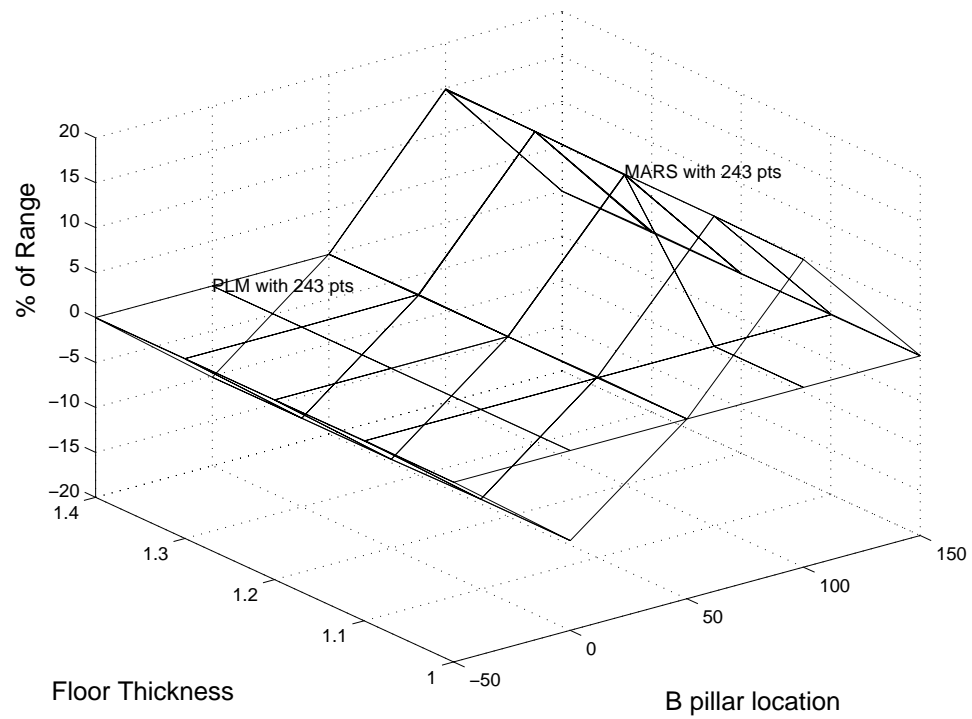


Figure 3.6: (ERMSE/response range) of polynomial and MARS models with 243 design points.

Table 3.2: Selected Numerical Results.

Model	No. of Design Points	Max Error	ERMSE at	
			Design Points	Test Points
LM	14	95.0222	4.0675	38.1969
QDM	14	96.5684	0.3146	37.8841
MARS	14	97.0071	26.1171	36.7300
LM	17	114.8618	8.2247	48.4791
MARS	17	-139.7960	14.3700	48.3807
LM	19	95.7740	5.1729	38.3018
QDM	19	97.0783	0.8729	37.8844
MARS	19	96.8069	24.0653	36.5292
LM	32	100.4263	0.0000	38.8571
HM	34	93.5924	0.1188	38.2424
HM	106	93.1696	0.0157	38.5856
LM	243	98.7086	6.5460	39.9432
QDM	243	92.1269	1.1224	39.4614
HM	243	93.1550	0.0000	39.4917
PLM	243	-51.3850	0.0000	7.2887
MARS	243	92.9333	0.2467	39.4931
LM	3125	77.9965	34.2793	N/A
QDM	3125	70.3151	33.7758	N/A
PLM	3125	0.0000	0.0000	N/A
MARS	3125	6.5911	1.4939	N/A

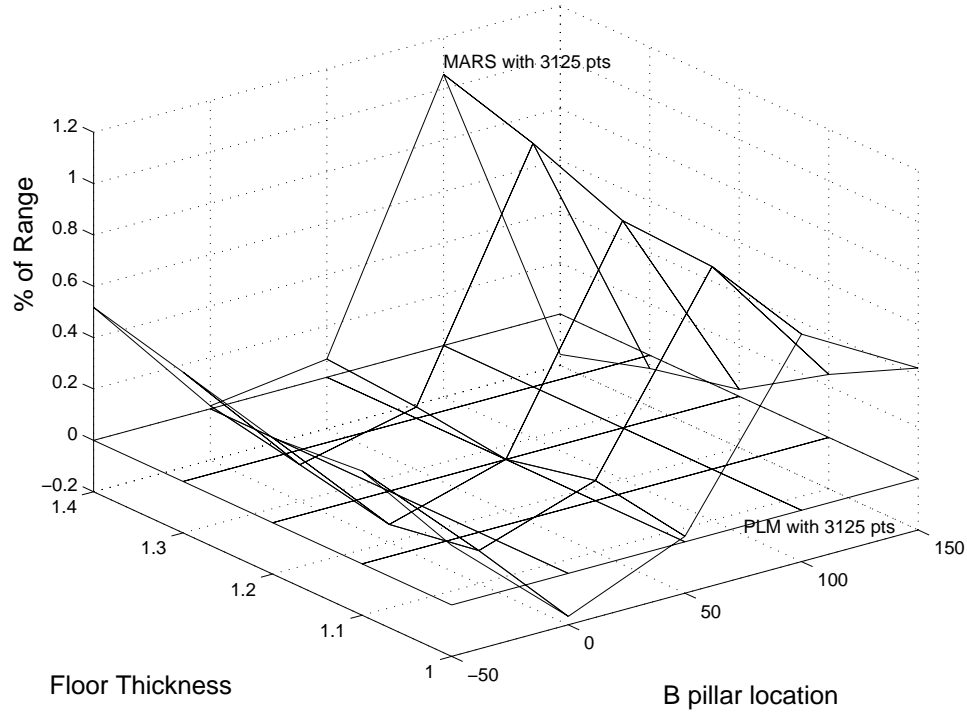


Figure 3.7: (ERMSE/response range) of polynomial and MARS models with 3125 design points.

model. Figure 3.10 shows the maximum error at all 3125 data points. Figures 3.9 and 3.10 illustrate how well each regression model approximates the data in areas of the design space away from the points used to build the regression model.

Finally, the error at the point where the bending stiffness itself is a maximum is shown in Table 3.3 and in Figure 3.11 for all models.

3.5.3 Base Functions Selected

When using the same set of design points, polynomial models can produce smaller empirical root-mean-square error (ERMSE) at all design points than MARS models, because the polynomial models are generated by the method of the least MSE. The ERMSE of all test points is a more important indicator of the quality of an approximation because it indicates how well the model fits the actual function at the unknown (for this model) points. It is used as an indication of the accuracy of each model. For MARS models, the accuracy is similar to the other models when the number of the points used to build the model is less than or equal to 243. The MSE for all test points is, of course, not meaningful when all 3125 points are used to generate the model (since there are no test points). Comparing the maximum error of each MARS model, the one with 3125 evaluations is much better

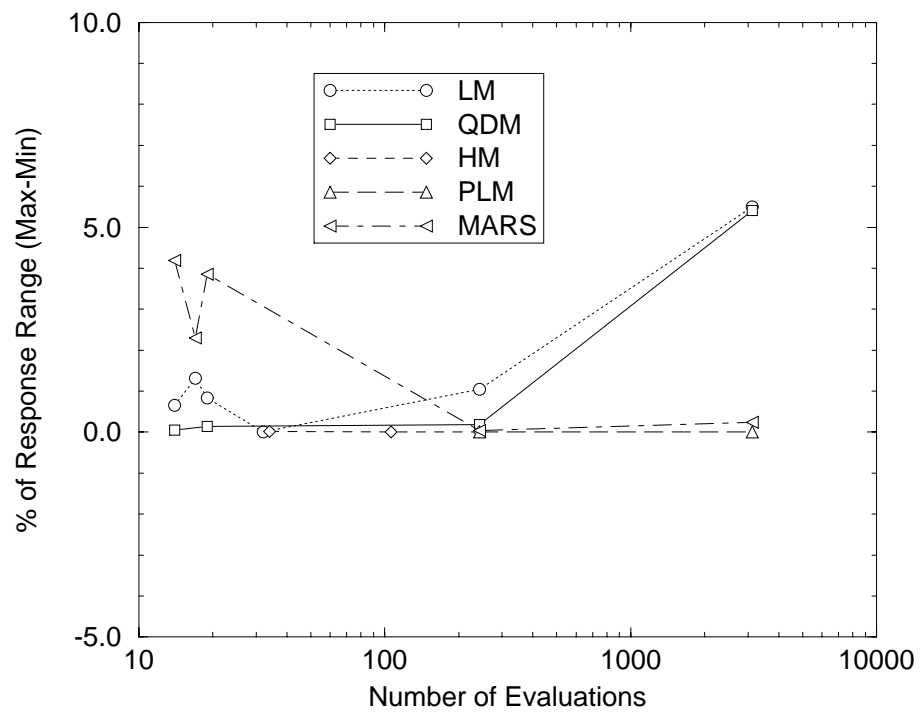


Figure 3.8: (ERMSE/response range) at all design points.

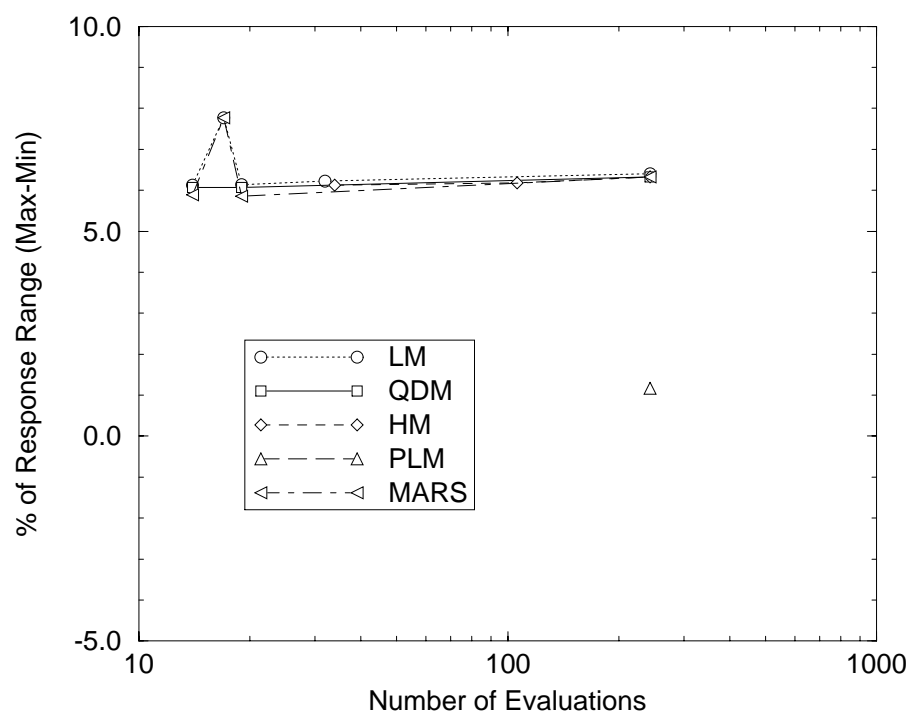


Figure 3.9: (ERMSE/response range) at all test points.

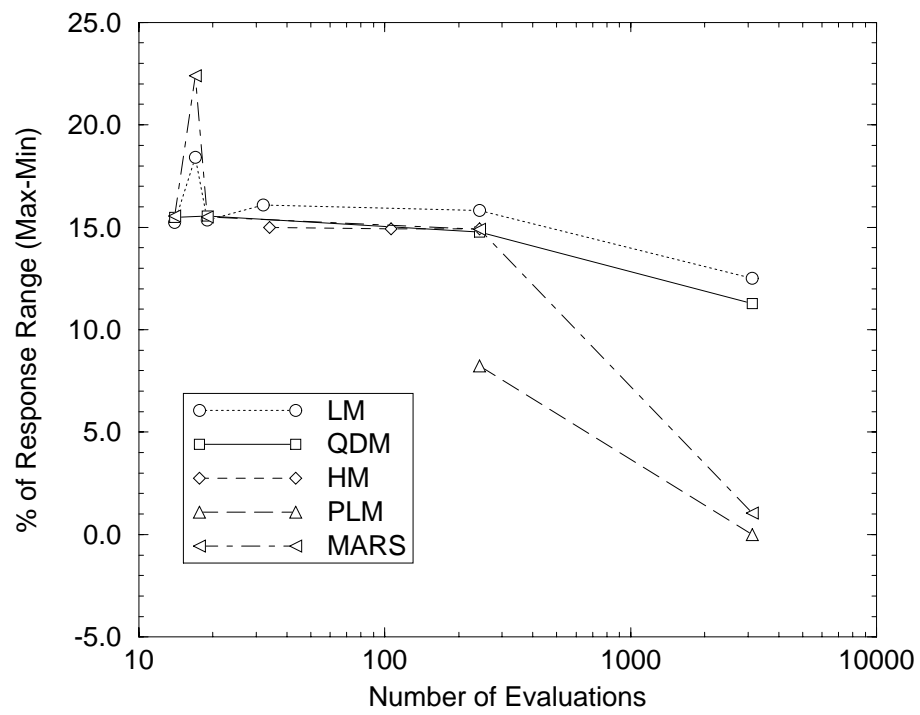


Figure 3.10: (maximum error/response range) of all design points and test points.

Table 3.3: Error at Maximum Response (3,364.9).

Model	No. of Design Points	Error at Max Response	Error/Max Response %
LM	14	92.5664	2.75
QDM	14	96.5684	2.87
MARS	14	93.9426	2.79
LM	17	111.5991	3.32
QDM	17	97.0783	2.89
MARS	17	130.299	3.87
LM	19	93.3192	2.77
QDM	19	97.0783	2.89
MARS	19	93.7131	2.78
LM	32	90.5300	2.69
HM	34	93.5924	2.78
HM	106	93.1696	2.77
LM	243	89.2554	2.65
QDM	243	93.1550	2.77
HM	243	93.1550	2.77
PLM	243	0.0000	0
MARS	243	92.9333	2.76
LM	3125	67.2762	2.00
QDM	3125	67.2762	2.00
PLM	3125	0.0000	0
MARS	3125	6.5911	0.20

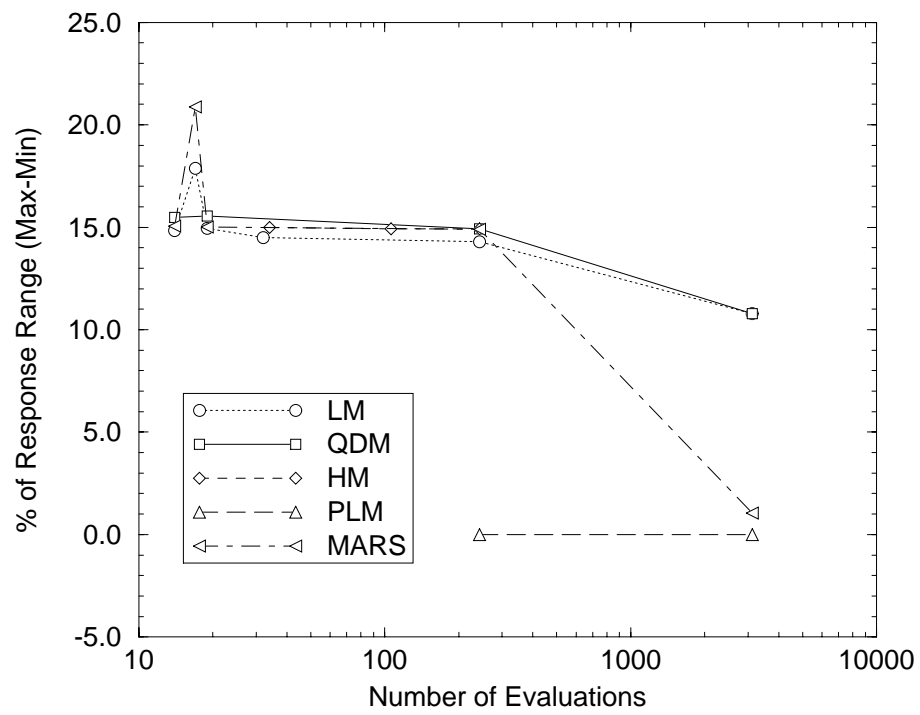


Figure 3.11: (error/response range) at maximum response.

than the one with 243 evaluations.

In the same way, similar conclusions can be drawn for polynomial models. The only difference is that the piecewise polynomial model with 243 evaluations is much better than the polynomial model with 32 evaluations. When the number of evaluations is small (≤ 32), there is no significant difference between the polynomial models and the MARS model. However, when the number of evaluations is increased and the piecewise linear model is used, the polynomial model is better than the MARS model, especially with 243 evaluations.

The single linear model with 3125 evaluations is also generated (listed in tables but not plotted in figures). The accuracy is almost the same as any other single polynomial model.

Among all the four measures, the ERMSE at design points is not important because those errors will be compensated by the second term of the interpolation model as in the Equation 3.3.

From the other three measures for all models with design points up to 243, the piecewise linear model (PLM) has the smallest maximum absolute error 51.385, the smallest ERMSE 7.2887, and the smallest error at the maximum response 0.0000. Although the error at the maximum response is meaningless because that point is included in 243 design points of PLM, it seems that the apparent choice should still be the piecewise linear model if the computational cost of the finite element model is negligible. But because the design of experiment for the PLM is a 3-level full factorial design with $3^n = 243$ design points, the PLM with $2^n = 32$ models is equivalent to dividing the DVS equally into 2^n sub-DVS and building one linear model for each sub-DVS. And if the cost of the computer experiment is not negligible, the number of design points should also be taken into consideration. For a ten-dimensional DVS ($n = 10$), the PLM needs $3^n = 59049$ design points, but the linear model or quadratic model will work with only 27 design points. So the PLM is not a feasible choice in practice. Actually, the good performance of the PLM only shows the effect of reducing the size or volume of the DVS to be exploited.

For all other models, the three measures only differ slightly. The high-order polynomial model and the MARS model, which includes nonlinear interactions, are not so appealing because these complex models could not outperform the simpler models. Because the two-way interactions can be transformed into second-order terms of main factors,

$$x'_1 \cdot x'_2 = (x_1 - x_2) \cdot (x_1 + x_2) = x_1^2 - x_2^2, \text{ if } x'_1 \triangleq (x_1 - x_2) \text{ and } x'_2 \triangleq (x_1 + x_2)$$

the quadratic model is balanced only with a CCD design based on a Resolution V fractional factorial

design, otherwise it is equivalent to a linear model with interaction terms under another coordinate system. The linear model will work well with a Resolution III fractional factorial design, which consists at most of half the design points needed for a balanced quadratic model.

Finally, based on a balance of all of these considerations, the linear model 3.8 is chosen as the base function.

3.6 Criteria for Sampling Design Points with *a priori* Information

Once the base functions and the cov structure of the interpolation model are decided, the metamodel can be built with a set of design points and the corresponding responses. When building the first metamodel at the first stage, in general it is not possible to assume any specific knowledge about the objective function because of the wide range of functions in engineering systems. So the initial design points are chosen by a experimental design method. After the first stage, some knowledge of the objective function is obtained from the responses at design points and the metamodel. It would be helpful if design points can be chosen based on data at previous stage or stages in order to build better metamodels.

After constructing the metamodel of the i^{th} stage, a set of n_i design points exists $S_i = \{s_1, \dots, s_{n_i}\}$. Now some new design points $\{s_{1+n_i}, \dots, s_{n_{1+i}}\}$ need to be chosen according to a sampling criterion, and append them to S_i in order to generate the $(i + 1)^{th}$ set of design points $S_{i+1} = \{s_1, \dots, s_{n_i}, s_{1+n_i}, \dots, s_{n_{1+i}}\}$.

Among many sampling criteria, the maximum entropy sampling criterion has a sound basis. Lindley [33] introduced ideas from Shannon's information theory into the area of experiment design, and established a measure of the information provided by an experiment. Later Shewry and Wynn [54] defined the maximum entropy sampling criterion for the set of fixed candidate design points.

For a random process Y in the design space, let $T = \{X_1, \dots, X_N\}$ be the set of all possible candidate design points and corresponding responses, and $S \subset T$ be the chosen design points, $\bar{S} = T \setminus S$ be the complementary set of S . The following relation can be obtained.

$$Ent(Y) = Ent(S) + E_S(Ent(\bar{S} | S)) \quad (3.15)$$

The second term of the right side is the entropy for the conditional distribution of the unsampled \bar{S} give chosen design points S . The natural Bayesian sampling criterion is to minimize it. But

because the left side, the entropy of the process, is fixed, minimizing the second term on the right side equals to maximizing the first term, which is the entropy of the chosen design points. If Y is a Gaussian process as assumed in Section 3.2, $Ent(Y)$ is, up to constants, $\log(\det(Cov(Y)))$. So the maximum entropy sampling criterion becomes to maximize the determinant of the posterior correlation matrix which is

$$D = [Cov(Y(s_i), Y(s_j))]_{1 \leq i \leq n_{i+1}, 1 \leq j \leq n_{i+1}} \quad (3.16)$$

The conclusions above are based on the assumption of finite set of possible design points. By the different use of the entropy formula, Wynn *et al.* proved the same conclusions while loosening the condition of discrete design points [53].

Also there are several variations of the maximum entropy criterion. If the first term in Equation 3.3 is a constant, then the criterion becomes maximizing $\det(D) \cdot \|D^{-1}\|$ [3]. If there is random error whose variance tends to ∞ in the model, the criterion becomes minimizing $\|D\|$ [36]. The operator $\|\cdot\|$ in above criteria is the sum of all entries in the covariance matrix.

Johnson *et al.* [22] proposed the *maximin distance design*. For the Gaussian process with the covariance structure as in Equation 3.4, the maximin distance design is asymptotically equivalent to the maximum entropy design under some conditions [22]. For any subset S of T containing n design points, S° is a maximin distance design if

$$\max_S \min_{s, s' \in S} d(s, s') = \min_{s, s' \in S^\circ} d(s, s') \quad (3.17)$$

where $d(s, s')$ is a distance function of a pair of design points. For the model in Section 3.2, $d(s, s')$ is the m -dimensional Euclidean distance. By maximizing the minimum distance between design points, this criterion tries to reduce the redundancy among design points.

Johnson *et al.* also proposed the *minimax distance design* in a similar way [22]. For any subset S of T containing n design points, we call S^* a minimax distance design if

$$\min_S \max_{t \in T} d(t, S) = \max_{t \in T} d(t, S^*) \quad (3.18)$$

where $d(t, S) = \min_{s \in S} d(t, s)$.

As in the case of the maximin distance design, for the Gaussian process with the covariance structure as in Equation 3.4, the minimax distance design is a *G-optimal* [23] design under some

conditions, where the *G-optimal* design will minimize the maximum variance of the fitted response over the design region as in the following equation:

$$\max_{t \in T} \text{var}(Y_t | Y_s, s \in S) / \text{var}(Y_t) \quad (3.19)$$

where Y is the random process, T is the finite set of all possible design points, and S is the chosen design points.

Following the induction of the interpolation model in Section 3.2, the integrated mean square error of the interpolation model, *IMSE*, can also be calculated:

$$IMSE = \frac{1}{\sigma_z^2 \Omega} \cdot \int_{X \in DVS} E_\theta[(Y(\hat{x}) - Y(x))^2] \cdot w(x) \cdot dX \quad (3.20)$$

where θ and σ_z^2 are parameters of the covariance structure of the random process, $w(x)$ is the weighting function, and $\Omega = \int_{X \in DVS} w(x) \cdot dX$.

Crary *et al.* [10] introduced I_z -Optimal designs which will minimize the *IMSE* of the interpolation model for computer experiments. The I_z -Optimal design can be generated by a program $I - OPT^{TM}$ which is available at <http://www-personal.engin.umich.edu/crary/iopt>. They also introduced the Bayesian *I-Optimality*, in which $X'WX$ is used to represent *a priori* information. Here X is the matrix of all design points, and W is the weight matrix. The Bayesian *I-Optimal* design can also be generated by the program $I - OPT^{TM}$.

All above criteria will be compared with random sampling, in which all coordinates of each design point are generated as uniformly distributed random numbers, in the tests described below.

3.7 Tests of Improvement between Metamodels

The metamodel using more design points would be expected to outperform the metamodel using fewer design points. A quantitative measure would be helpful to know how good the performance improvement is. To compare two metamodels, one method is to compute the errors of the metamodel over a grid in DVS. This method will be used to compare different sampling criteria. The disadvantage of this method is that it requires lots of evaluations of the complex software which is just opposite of the purpose of using metamodels, so it is crucial to find a way to compare metamodels at different stages using a limited number of evaluations.

Consider the comparison between the metamodel at stage i and the metamodel at stage j , *i.e.*,

$\hat{Y}_i(x)$ and $\hat{Y}_j(x)$. Without loss of generality, assume $i > j$. A new set of design points is needed to compare $\hat{Y}_i(x)$ and $\hat{Y}_j(x)$, and design points $\{s_{n_i+1}, \dots, s_{n_{i+1}}\}$ are needed to construct $\hat{Y}_{i+1}(x)$. So it is convenient to use $\{s_{n_i+1}, \dots, s_{n_{i+1}}\}$ to test $\hat{Y}_i(x)$ and $\hat{Y}_j(x)$. Let $k = n_{i+1} - n_i$, then two groups of errors, $\{e_{i,1}, \dots, e_{i,k}\}$ for $\hat{Y}_i(x)$ and $\{e_{j,1}, \dots, e_{j,k}\}$ for $\hat{Y}_j(x)$, can be obtained.

Repeated computer experiments with the same parameters generate same results, so the error for a metamodel at any specific point in the DVS is the same if the metamodel is compared with computer experiments repeatedly at the same point. But if the overall performance of the metamodel in the DVS is of interest but not the performance at any specific design point, then errors of the metamodel in the whole DVS can be considered as a population with certain probability distribution. With this, some of the techniques in traditional experiment analysis can be applied to the metamodel of the computer experiment.

3.7.1 Test with Assumption about the Distribution of the Error

According to the central limit theorem, the sum of n identically distributed random variables has an approximate normal distribution. Furthermore, the Liapunov theorem states that the sum of n random variables with different means and variances still has an approximate Normal distribution if some conditions are satisfied. If the error is considered as the sum of many disturbances among which there is no overwhelming factor, then the error random variable can be assumed to be Normal.

Let E_i be the random variable for the error of $\hat{Y}_i(x)$, and E_j be the random variable for the error of $\hat{Y}_j(x)$. Then under the assumption of their distribution, $E_i \sim N(\mu_i, \sigma_i^2)$ and $E_j \sim N(\mu_j, \sigma_j^2)$. The sample means and sample variances of E_i and E_j are

$$\begin{aligned}\bar{e}_i &= \frac{1}{k} \sum_{l=1}^k e_{i,l} \\ \bar{e}_j &= \frac{1}{k} \sum_{l=1}^k e_{j,l} \\ S_i^2 &= \frac{1}{k-1} \sum_{l=1}^k e_{i,l}^2 - k \cdot \bar{e}_i^2 \\ S_j^2 &= \frac{1}{k-1} \sum_{l=1}^k e_{j,l}^2 - k \cdot \bar{e}_j^2\end{aligned}\tag{3.21}$$

If μ_i and μ_j are to be compared, consider the statistic

$$T = \frac{(\bar{e}_j - \bar{e}_i) - (\mu_j - \mu_i)}{\sqrt{\frac{S_j^2}{k} + \frac{S_i^2}{k}}} \quad (3.22)$$

Let the null hypothesis and alternative hypothesis be

$$\begin{aligned} H_0 : \mu_j - \mu_i &= 0 \\ H_1 : \mu_j - \mu_i &> 0 \end{aligned} \quad (3.23)$$

Under H_0

$$T' = \frac{(\bar{e}_j - \bar{e}_i)}{\sqrt{\frac{S_j^2}{k} + \frac{S_i^2}{k}}} \sim t_{k-1} \quad (3.24)$$

H_0 will be rejected if $T' > t_{k-1, 1-\alpha}$ with significance level of α , which means that the probability of erroneously accepting $H_1 : \mu_j - \mu_i > 0$ is α . Here the value of T' is computed, then the smallest α is chosen which will cause the rejection of H_0 .

It may be useful to compare σ_i^2 and σ_j^2 , the variances of the two random variables. Then consider another statistic

$$F_0 = \frac{(k-1) \cdot S_j^2 / \sigma_j^2}{(k-1) \cdot S_i^2 / \sigma_i^2} = \frac{S_j^2}{S_i^2} \cdot \frac{\sigma_i^2}{\sigma_j^2} \quad (3.25)$$

and make the following null hypothesis and alternative hypothesis:

$$\begin{aligned} H_0 : \sigma_j^2 &= \sigma_i^2 \\ H_1 : \sigma_j^2 &> \sigma_i^2 \end{aligned} \quad (3.26)$$

Under H_0

$$F'_0 = \frac{S_j^2}{S_i^2} \sim F_{k-1, k-1} \quad (3.27)$$

H_0 will be rejected if $F_0 > F_{k-1, k-1, 1-\alpha}$ with significance level of α , which means that the probability of erroneously accepting $H_1 : \sigma_j^2 > \sigma_i^2$ is α . Here the value of F_0 is computed first, then the smallest α is chosen which will cause the rejection of H_0 .

3.7.2 Test without Assumption about the Distribution of the Error

The tests in previous section are based on the assumption that the distribution of the error random variable is Normal. Sometimes this assumption is valid, but in general the distribution of the error

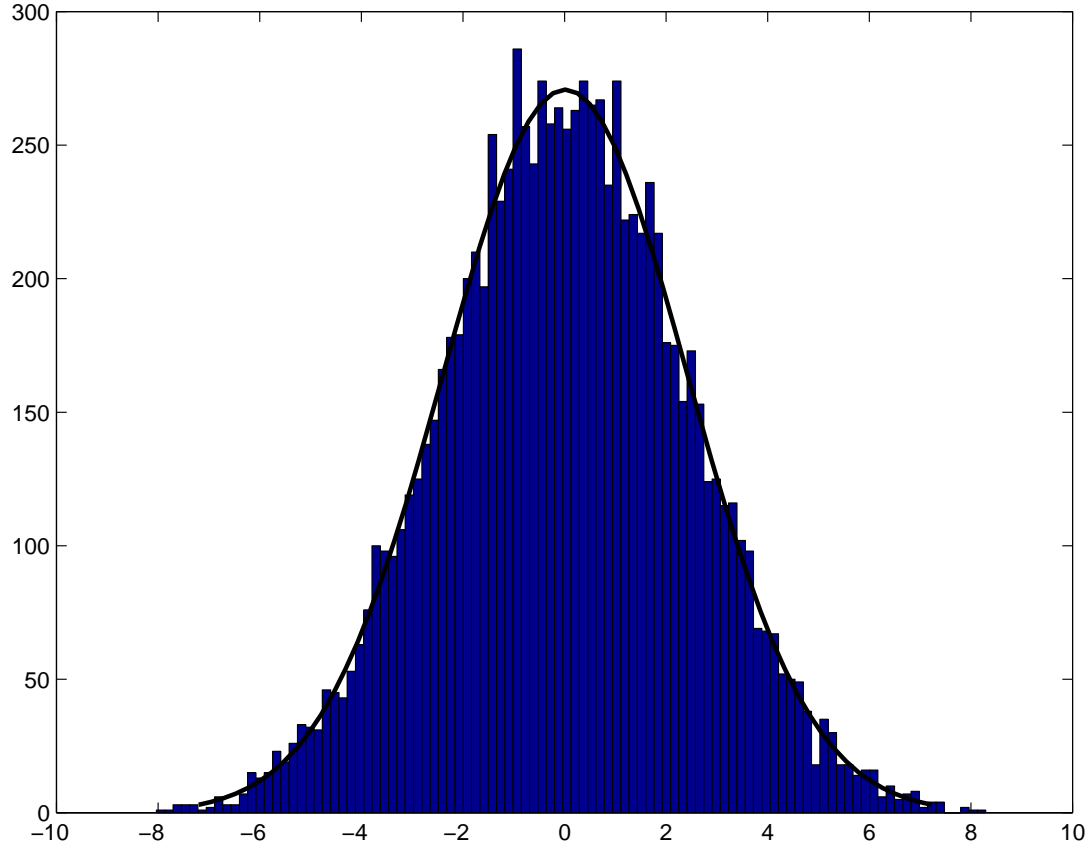


Figure 3.12: Histogram with fitted normal density of e_1 at 10,000 random test points.

is unknown or its distribution is known to be not Normal. For example, consider two random error variables: $e_1 = x_1^3 + x_2 + \dots + x_6$ and $e_2 = 10 \cdot x_1^3 + x_2 + \dots + x_6$. The only difference between e_1 and e_2 is the weights on x_1^3 are 1 and 10. Figure 3.12 and Figure 3.13 are histograms with fitted Normal density of e_1 and e_2 at 10,000 uniformly random test points (The 10,000 test points used here and later are only used to test the Normality assumption. They are not available to construct metamodels). As is apparent, the Normality assumption is a good approximation for probability density function (PDF) for e_1 , but not for e_2 .

Under either circumstance, if the test points $\{s_{n_i+1}, \dots, s_{n_{i+1}}\}$ are randomly generated, tests similar to those in Section 3.7.1 can be carried out with a different method. This method is called randomization test [6]. When two random variables are considered equivalent in the sense of any numerical characteristics, switching any pair of samples from each random variable will not affect corresponding statistics.

To compare μ_i and μ_j , consider the statistic:

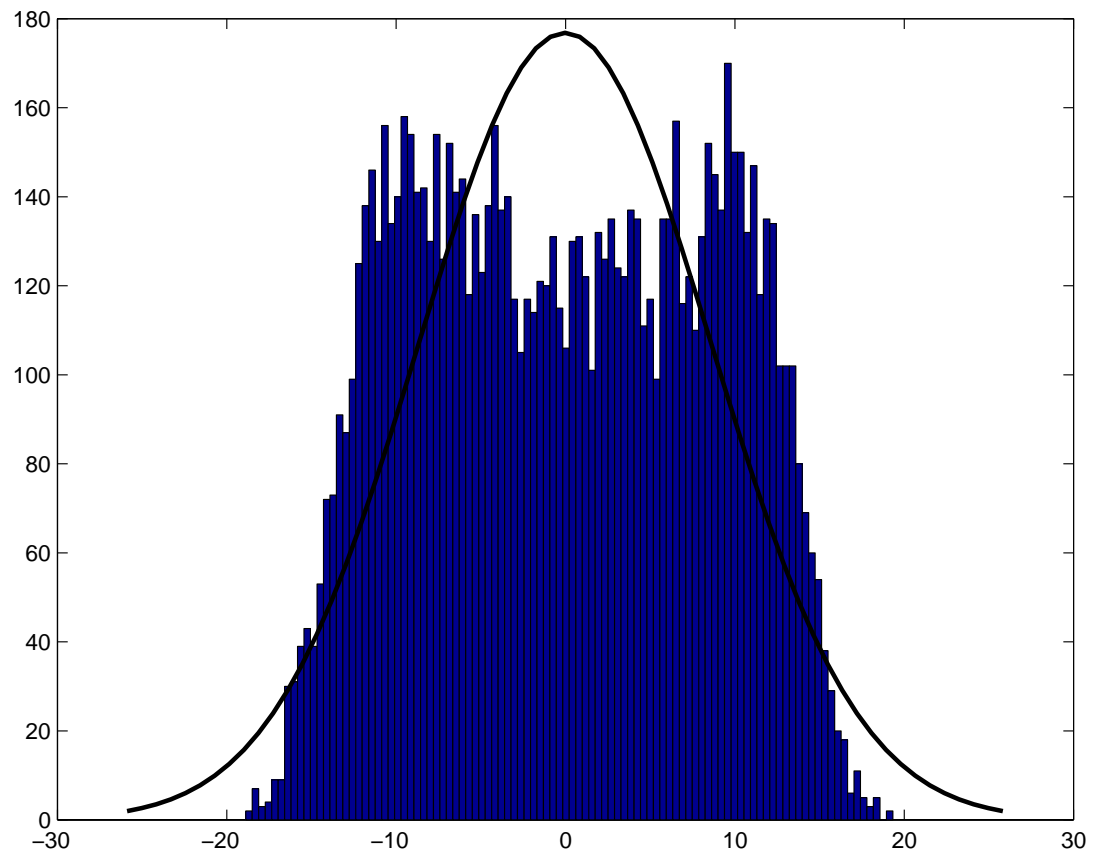


Figure 3.13: Histogram with fitted normal density of e_2 at 10,000 random test points.

$$\overline{de}_0 = \frac{1}{k} \cdot \sum_{l=1}^k (e_{j,l} - e_{i,l}) \quad (3.28)$$

If some pairs of $e_{i,l}$ and $e_{j,l}$ are exchanged in Equation 3.28, 2^k possible results can be obtained. The same hypotheses as those in 3.23 are made. Under H_0 there is no difference between $e_{i,l}$ and $e_{j,l}$ in Equation 3.28 because they are values of the random variables with the same mean value at randomly generated design points. So all these 2^k possible results are equally likely. Compare \overline{de}_0 with all 2^k results, suppose that m_1 results are larger than \overline{de}_0 and m_2 results are equal to \overline{de}_0 .

H_0 will be rejected at significance level of α , which means that the probability of erroneously accepting $H_1 : \mu_j > \mu_i$ is α , where

$$\alpha = \frac{m_1 + \frac{m_2}{2}}{2^k} \quad (3.29)$$

To compare σ_i^2 and σ_j^2 , consider the set $\{e_{j,1}, \dots, e_{j,k}, e_{i,1}, \dots, e_{i,k}\}$, which consists of $2k$ errors. There are $m = \frac{(2 \cdot k)!}{k!^2}$ possible ways to separate the $2k$ errors into two sets with the same size k . The F' for each possible partition of the $2k$ errors can be computed using Equation 3.27 and Equation 3.21. The hypotheses are the same as those in Equation 3.26. Under H_0 , there is no difference in the result of sample variance if any pair of elements in two partitions of the set of $2k$ errors are switched because they are values of two random variables with the same variance at randomly generated design points. So the sample variances of all m partitions are equally likely. Compare F_0 with all m possible F s, it can be seen that m_1 possible F s are larger than F_0 , and m_2 possible F s are equal to F_0 . H_0 will be rejected at significance level of α , which means that the probability of erroneously accepting $H_1 : \sigma_j^2 > \sigma_i^2$ is α , where

$$\alpha = \frac{m_1 + \frac{m_2}{2}}{\frac{(2 \cdot k)!}{(k!)^2}} = \frac{(m_1 + \frac{m_2}{2}) \cdot (k!)^2}{(2 \cdot k)!} \quad (3.30)$$

3.8 First Examples and Results

The first example function to be fitted is a ten-dimensional function

$$\begin{aligned} Y_{10d} = & \alpha_0 + \sum_{i=1}^{10} [\alpha_i x_i + \beta'_i \sin(\beta''_i x_i + \beta'''_i) + \gamma'_i \exp(\gamma''_i x_i + \gamma'''_i)] \\ & + \sum_{i=1}^{10} \sum_{j=i}^{10} [\alpha_{i,j} x_i x_j + \beta'_{i,j} \sin(\beta''_{i,j} x_i x_j + \beta'''_{i,j}) + \gamma'_{i,j} \exp(\gamma''_{i,j} x_i x_j + \gamma'''_{i,j})] \end{aligned}$$

Table 3.4: Ratio of ERMSE to Response Range of the 10-D Function.

Sampling Criterion	Number of Design Points			
	16	32	48	64
Entropy Criterion	13.8424%	10.7751%	8.5945%	6.8520%
Maximin Criterion	13.8424%	11.4704%	10.1075%	9.4655%
Random Method	13.8424%	12.6506%	10.9997%	10.5827%
Bayesian I-Optimal	13.8424%	8.7966%	7.0228%	6.9093%

$$\begin{aligned}
& + \sum_{i=1}^{10} \sum_{j=i}^{10} \sum_{k=j}^{10} [\alpha_{i,j,k} x_i x_j x_k + \beta'_{i,j,k} \sin(\beta''_{i,j,k} x_i x_j x_k + \beta'''_{i,j,k}) \\
& + \gamma'_{i,j,k} \exp(\gamma''_{i,j,k} x_i x_j x_k + \gamma'''_{i,j,k})]
\end{aligned} \tag{3.31}$$

All the α parameters are random numbers from $[-16, 16]$. All the β and γ parameters are random numbers from $[-2, 2]$.

The initial design of experiments is a Resolution III fractional factorial design with 16 corner points in the design space. Then based on each sampling criterion, three more metamodels will be built with 16 more design points for each metamodel.

Each metamodel will be tested at the 5^{10} points on the grid $[-1, -0.5, 0, 0.5, 1]^{10}$, and the ERMSE is computed. Figures 3.14 and Table 3.4 contain the results of the average errors for all metamodels. The results of maximum errors for all metamodels are in Figures 3.15 and Table 3.5.

It can be seen that the resulting design points by all four sampling criteria improve the accuracy of the metamodel. By adding 16 more design points, the ratio of the ERMSE to the response range decreases by about 2%, and the ratio of the maximum error to the response range decreases by roughly about 8%. The effect of adding 16 more design points decreased when the total number of design points increases. Among all four sampling criteria, it can be seen that the Bayesian I-Optimal design outperforms the other three criteria with 32 and 48 design points, and it is only second to the result of the entropy criterion with 64 design points.

For the metamodels based on the random sampling criterion, significance levels of tests about mean and variance are computed with the two methods described in Section 3.7. The results are shown in in Table 3.6. The resulting significance levels indicate that the probability of erroneously accepting $\mu_i > \mu_{i+1}$ is low, and the probability of erroneously accepting $\sigma_i^2 > \sigma_{i+1}^2$ is also low. By

Table 3.5: Ratio of Maximum Error to Response Range of the 10-D Function.

Sampling Criterion	Number of Design Points			
	16	32	48	64
Entropy Criterion	73.5424%	64.6439%	55.1691%	48.5776%
Maximin Criterion	73.5424%	64.1262%	61.4853%	59.3365%
Random Method	73.5424%	70.3027%	63.7913%	62.7712%
Bayesian I-Optimal	73.5424%	55.9763%	51.6001%	51.9713%

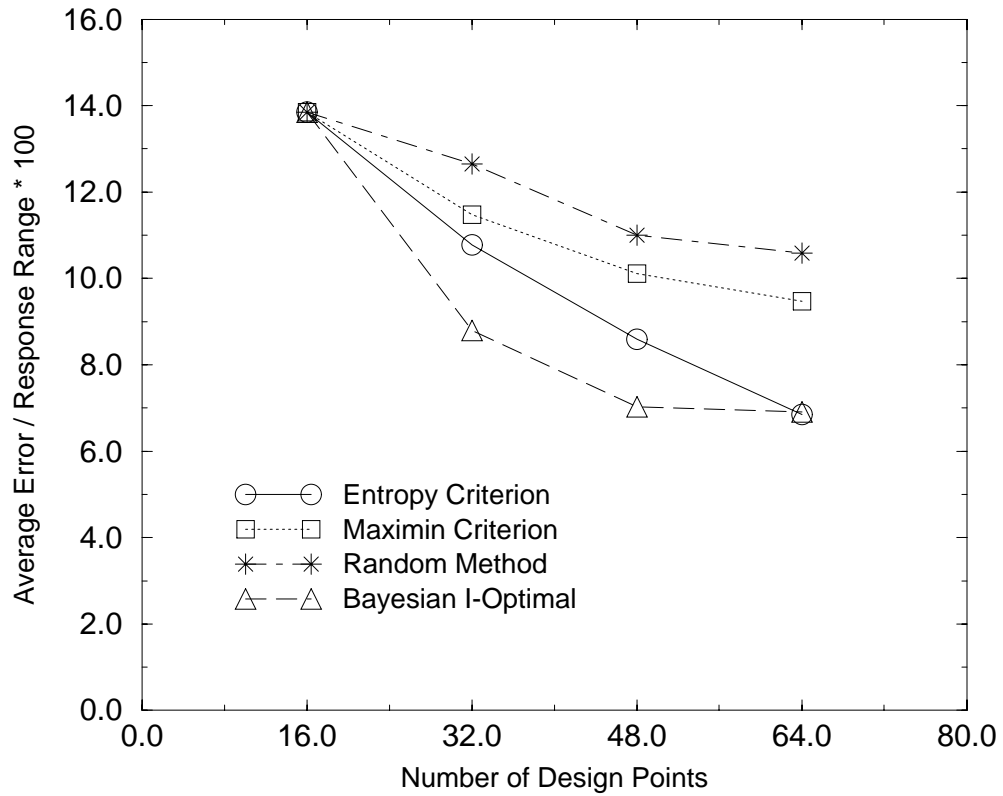


Figure 3.14: Ratio of ERMSE to response range of the 10-D function.

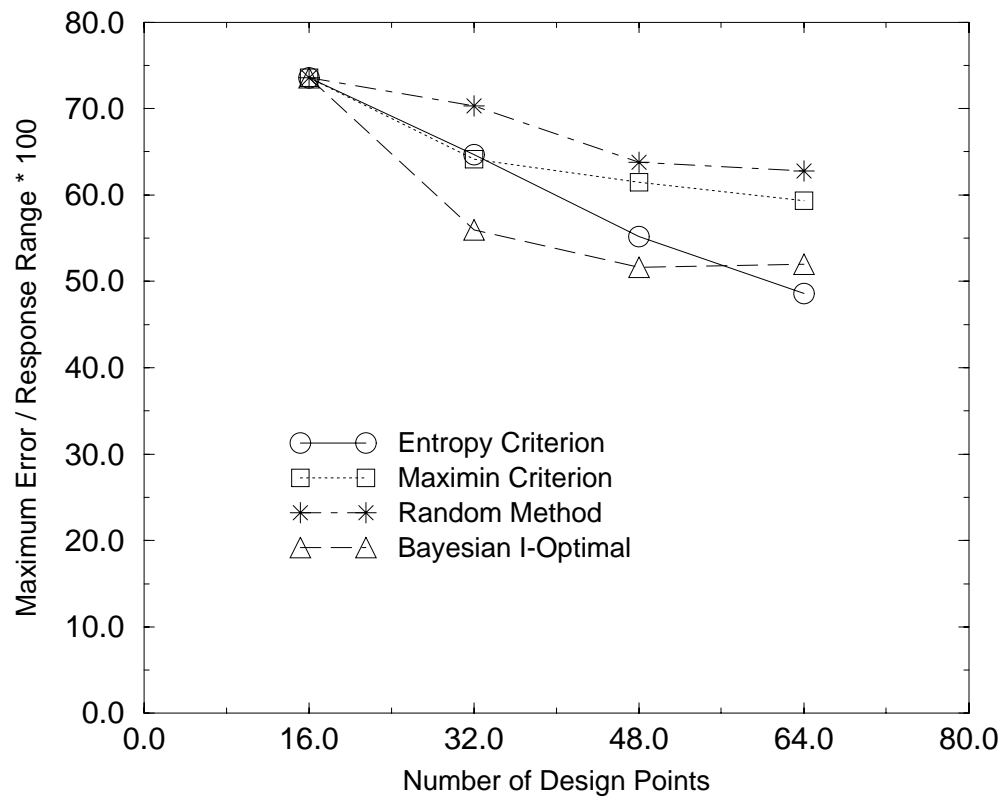


Figure 3.15: Ratio of maximum error to response range of the 10-D function.

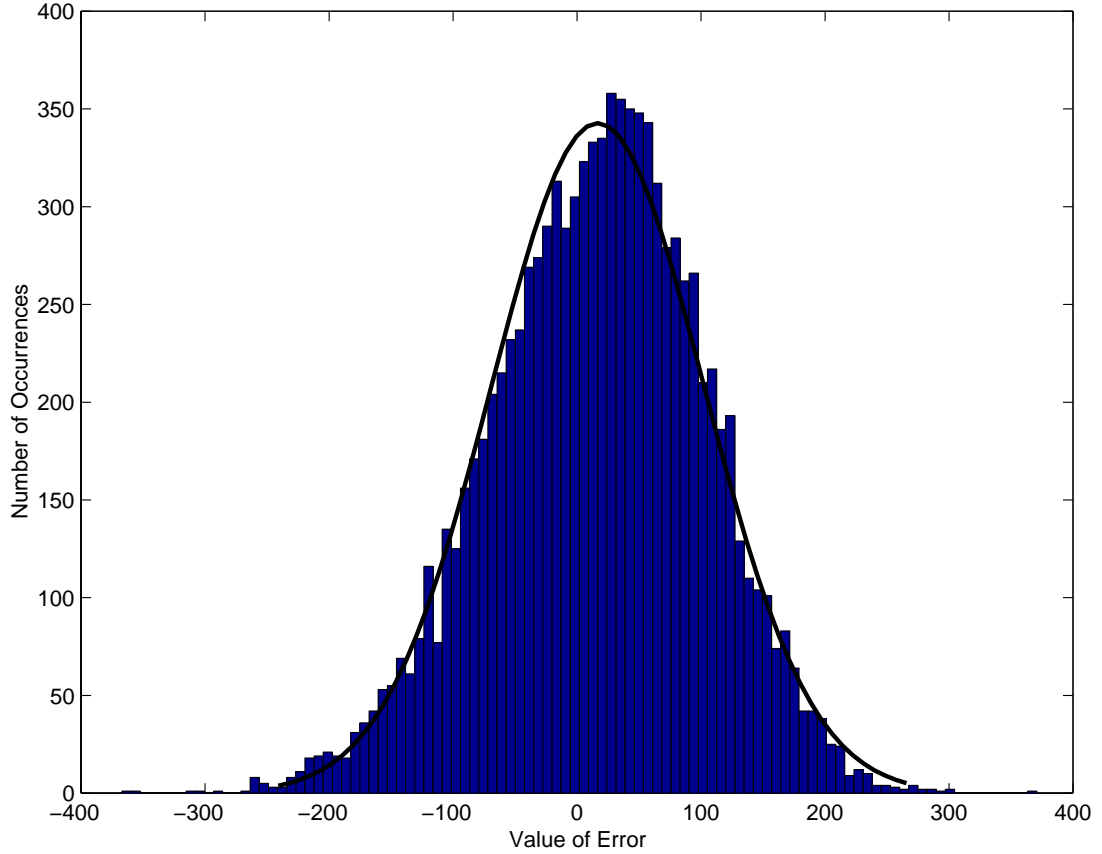


Figure 3.16: Histogram with fitted Normal density for the errors of \hat{Y}_1 of the 10-D function.

comparing the results from Normality tests and randomization tests, it can be seen that both tests generate similar results except the test of μ 's for the first and the second metamodels.

The histogram with superimposed fitted Normal density of errors of \hat{Y}_1 , \hat{Y}_2 and \hat{Y}_3 , where \hat{Y}_k is the k^{th} metamodel constructed on the k^{th} set of design points S_k , is presented in Figures 3.16, 3.18, and 3.20, respectively. The Normal probability plots (NPP) for these errors are shown in Figures 3.17, 3.19, and 3.21. From both the histograms and NPP of these errors, it can be seen that the Normality assumption is valid for the metamodels of ten-dimensional test function.

3.9 Second Examples and Results

The second test function is the bending stiffness of a Volkswagen passenger automobile chassis (shown in Figure 3.2) computed from a finite-element model (shown in Figure 3.3) in a design

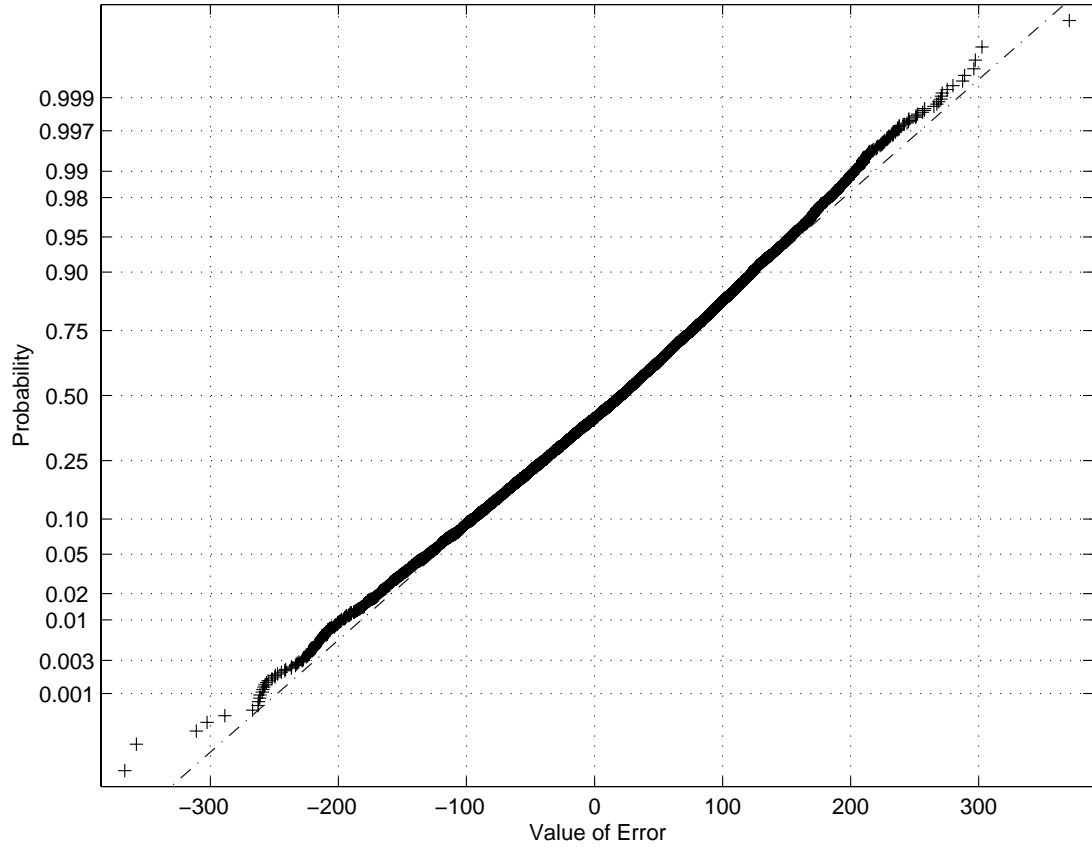


Figure 3.17: Normal probability plot for the errors of \hat{Y}_1 of the 10-D function.

Table 3.6: Significance Levels of Different Tests on the 10-D Function.

Tested Models	Normality Tests		Randomization Tests	
	t	F	t	F
\hat{Y}_1, \hat{Y}_2 on $\{s_{33}, \dots, s_{48}\}$	21.0153	48.8245	1.5343	48.6295
\hat{Y}_2, \hat{Y}_3 on $\{s_{49}, \dots, s_{64}\}$	29.2422	27.5828	5.04532	29.7665

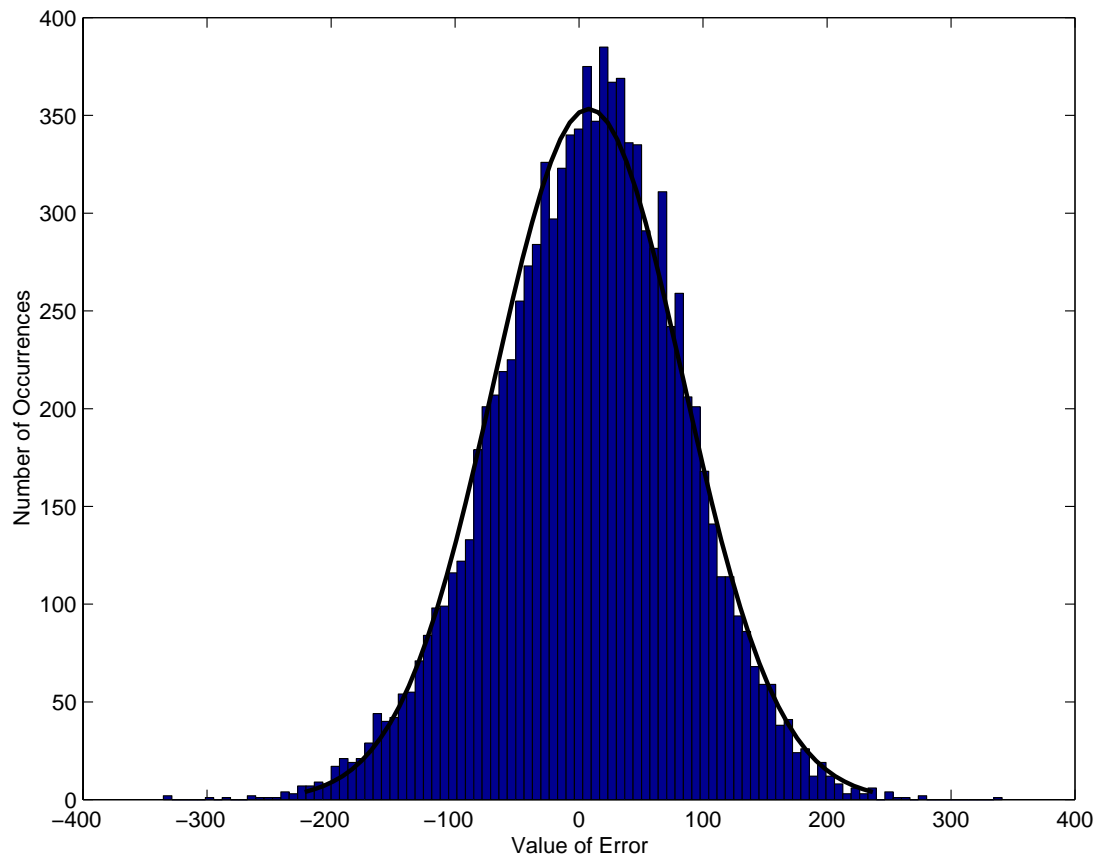


Figure 3.18: Histogram with fitted Normal density for the errors of \hat{Y}_2 of the 10-D function.

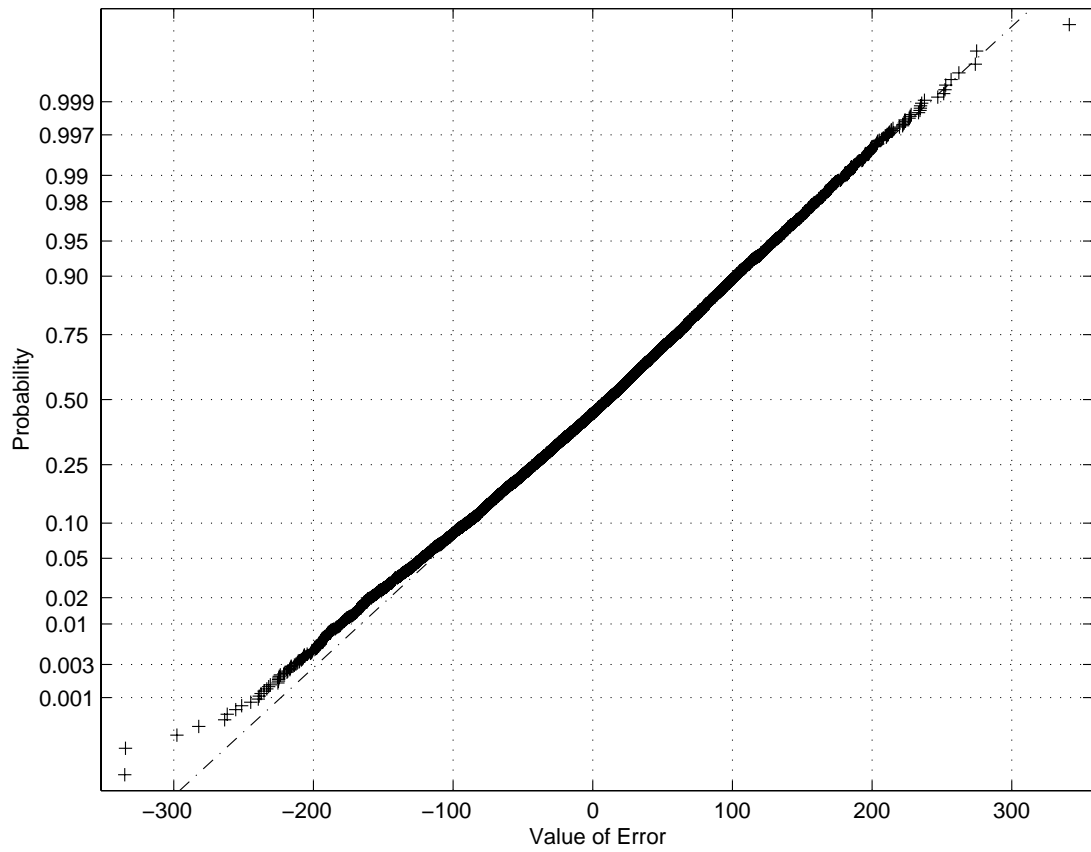


Figure 3.19: Normal probability plot for the errors of \hat{Y}_2 of the 10-D function.

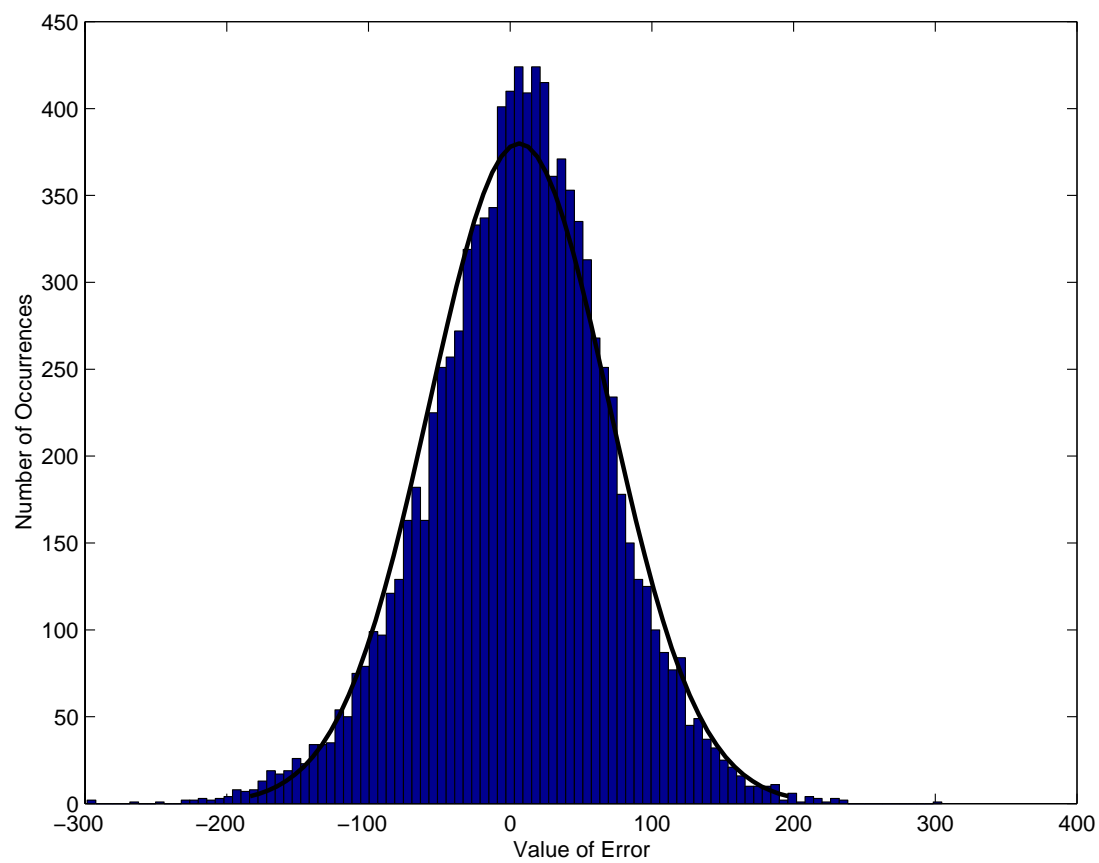


Figure 3.20: Histogram with fitted Normal density for the errors of \hat{Y}_3 of the 10-D function.

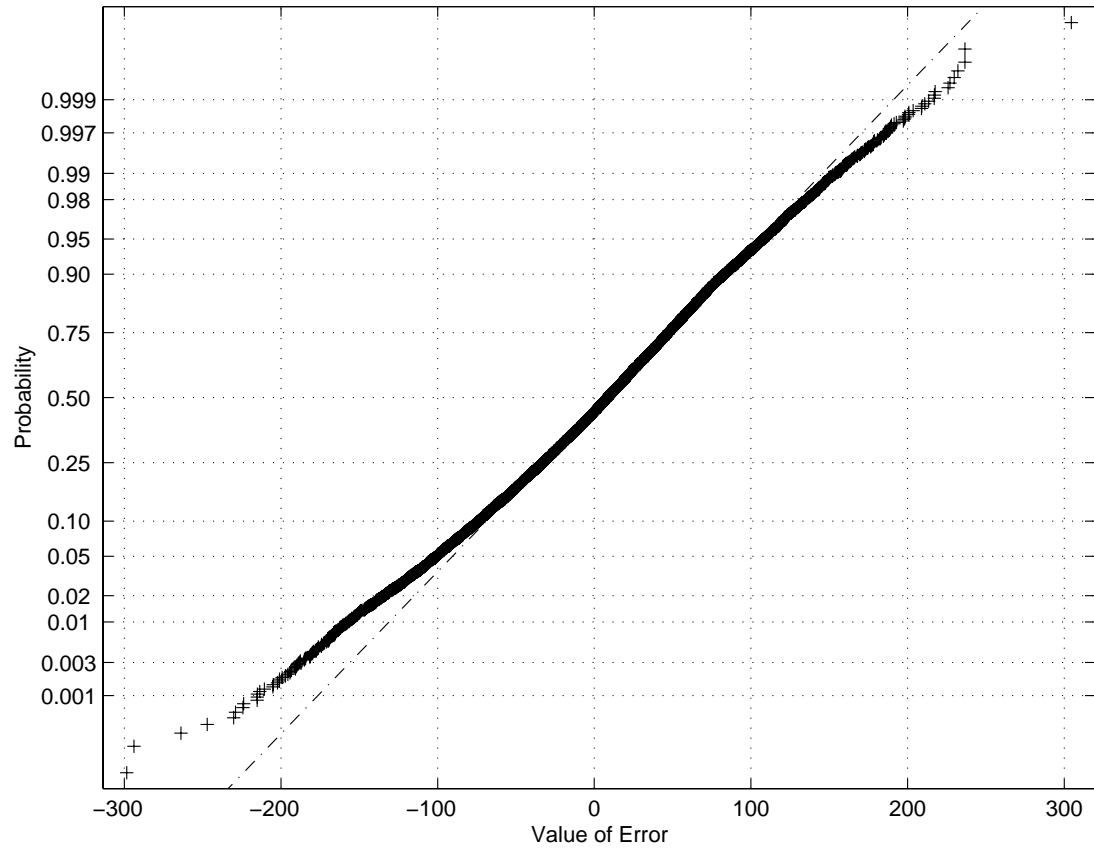


Figure 3.21: Normal probability plot for the errors of \hat{Y}_3 of the 10-D function.

space of five variables ($d = 5$):

$$\begin{aligned}
 x_1 &= \text{A Pillar Thickness [mm]} \\
 x_2 &= \text{B Pillar Thickness [mm]} \\
 x_3 &= \text{Floor Rail Thickness [mm]} \\
 x_4 &= \text{Floor Thickness [mm]} \\
 x_5 &= \text{B Pillar Location [mm]}
 \end{aligned}$$

Each design variable has been standardized to $[-1, 1]$ before constructing the metamodel.

The initial design of experiment is a Resolution III fractional factorial design with 8 corner points in the design space. Then based on each sampling criterion, three more metamodels will be built with 16 more design points for each metamodel.

Each metamodel will be tested at the 5^5 points on the grid $[-1, -0.5, 0, 0.5, 1]^5$, and the ERMSE is computed. Figure 3.22 and Table 3.7 contain the results of the average errors for all metamodels. The results of maximum errors for all metamodels are in Figure 3.23 and Table 3.8.

It can be seen that the resulting design points by all four sampling criteria improve the accuracy of the metamodel. By adding 16 more design points, the ratio of the ERMSE to the response range decreases by about 0.5%, and the ration of the maximum error to the response range decreases by roughly about 0.5%. The effect of adding 16 more design points decreased when the total number of design points increases. Among all four sampling criteria, it can be seen that the entropy criterion outperforms the other three in the ERMSE, and the random method is the best one in the maximum error. It should be noticed that the Bayesian I-Optimal design has almost no effect in both ERMSE and the maximum error.

For the metamodels based on the random sampling criterion, significance levels of tests about mean and variance are computed with two methods described in Section 3.7. The results are shown in Table 3.9. The resulting significance levels indicate that the probability of erroneously accepting $\mu_i > \mu_{i+1}$ is low, and the probability of erroneously accepting $\sigma_i^2 > \sigma_{i+1}^2$ is also low. By comparing the results from Normality tests and randomization tests, it can be seen that both tests generate similar results.

The histograms with superimposed fitted Normal density of errors of \hat{Y}_1 , \hat{Y}_2 , and \hat{Y}_3 , are presented in Figures 3.24, 3.26, and 3.28, respectively. The Normal probability plots (NPP) are shown in Figures 3.25, 3.27, and 3.29. From both the histograms and NPP of these errors, it can be seen

Table 3.7: Ratio of ERMSE to Response Range of the VW Model.

Sampling Criterion	Number of Design Points			
	8	16	24	32
Entropy Criterion	6.2127%	5.4689%	5.0754%	4.5471%
Maximin Criterion	6.2127%	5.5805%	5.2125%	4.9149%
Random Method	6.2127%	5.8470%	5.4753%	5.3617%
Bayesian I-Optimal	6.2127%	6.2391%	6.1680%	6.1845%

Table 3.8: Ratio of Maximum Error to Response Range of the VW Model.

Sampling Criterion	Number of Design Points			
	8	16	24	32
Entropy Criterion	15.5695%	16.5687%	15.0072%	13.4118%
Maximin Criterion	15.5695%	16.4573%	15.2478%	14.1349%
Random Method	15.5695%	14.4502%	13.2244%	12.9143%
Bayesian I-Optimal	15.5695%	15.6412%	15.8285%	15.8930%

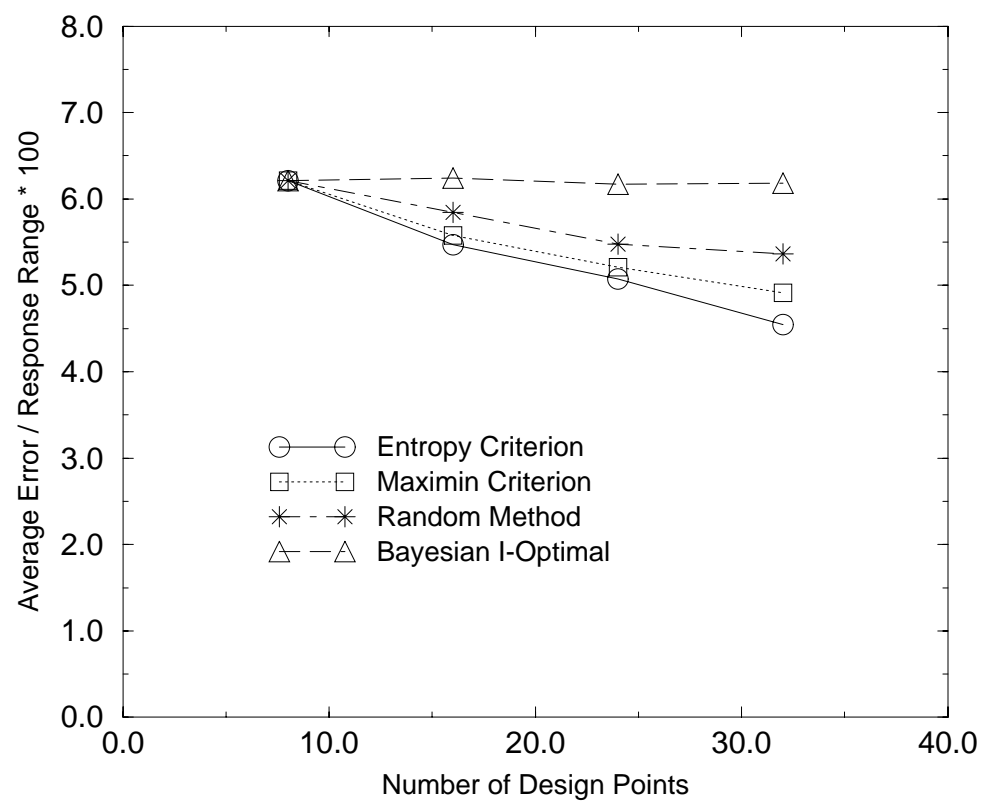


Figure 3.22: Ratio of ERMSE to response range of the VW model.

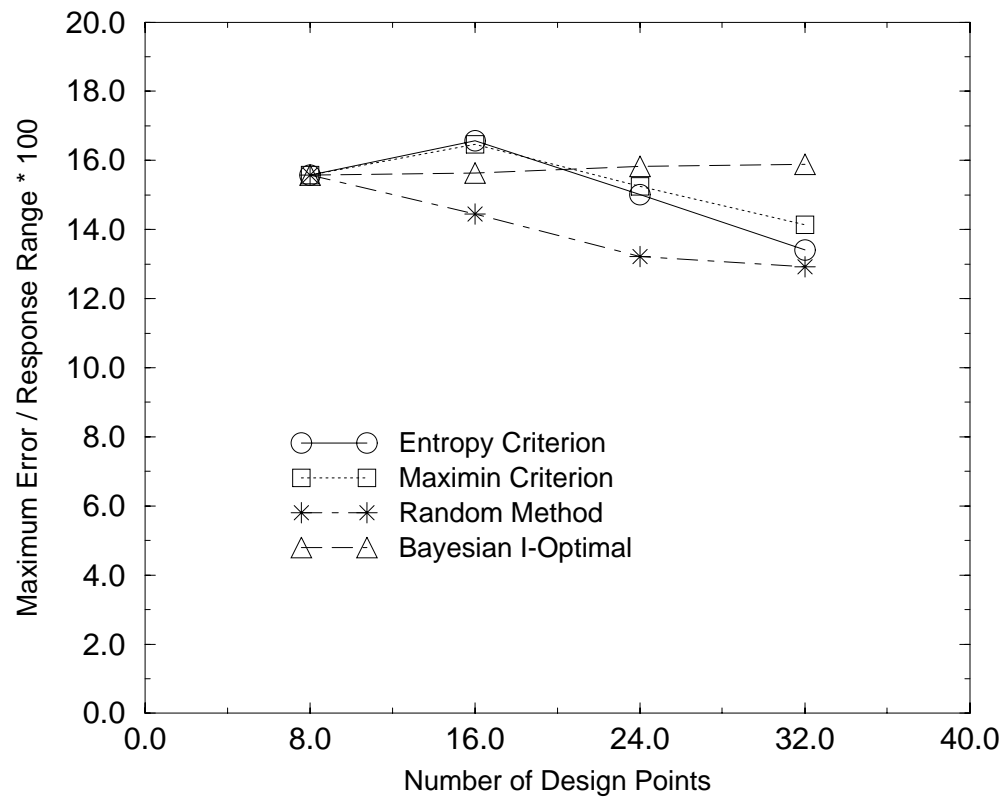


Figure 3.23: Ratio of maximum error to response range of the VW model.

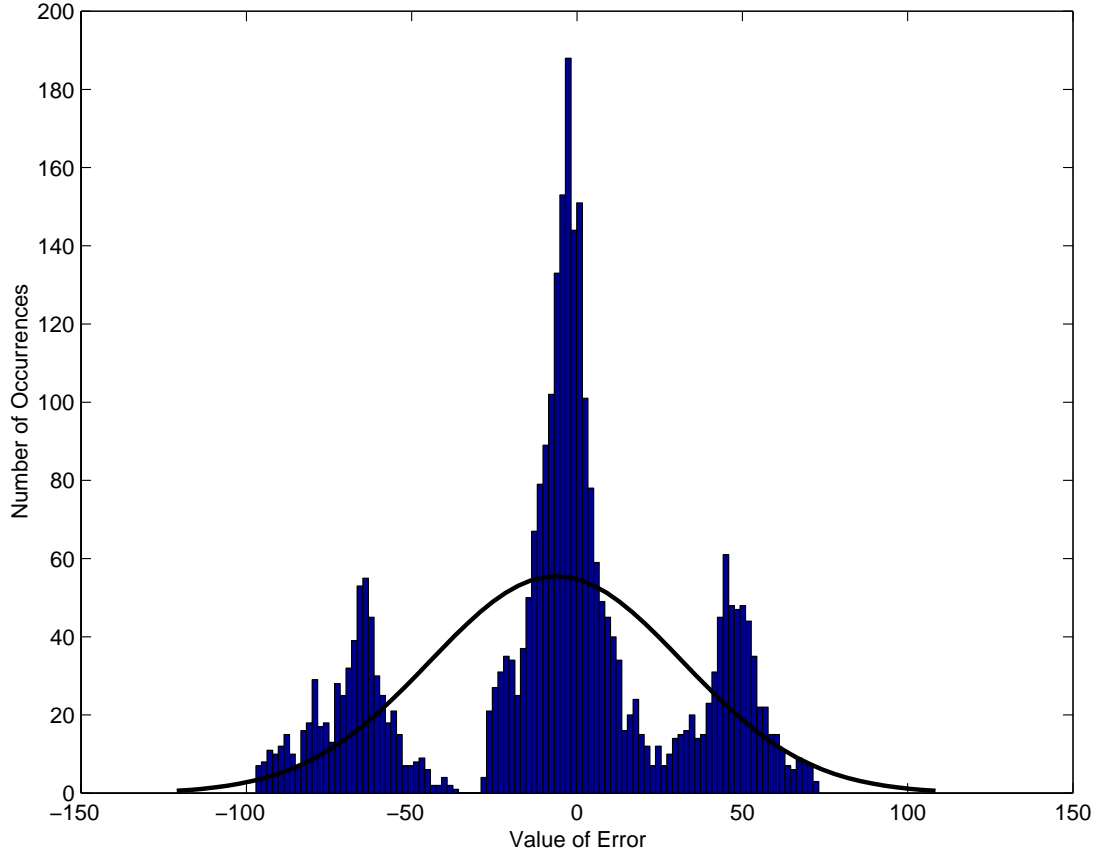


Figure 3.24: Histogram with fitted Normal density for the errors of \hat{Y}_1 of the VW model.

that the Normality assumption is invalid for the metamodels of the VW model.

3.10 Discussions and Conclusions

Because resources are limited, sometimes a metamodel must be used instead of a complicated simulation software package. Two of the most important factors to consider are the metamodel structure and the sampling criterion if multistage metamodels are needed. Experimental design is also important to increase the efficiency of the design points, but it is strongly dependent on the structure of the metamodel. A reliable measure of the performance improvement of the metamodel is also helpful to increase confidence about metamodels.

There are several metamodel structures for the deterministic computer experiments. Among them, the model proposed by J. Sacks *et al.* is the most widely used and is also suitable for the functions in many engineering design problems. This model consists of two parts, a approximation and a probability model of the approximation error. Sometimes the approximation is only a constant,

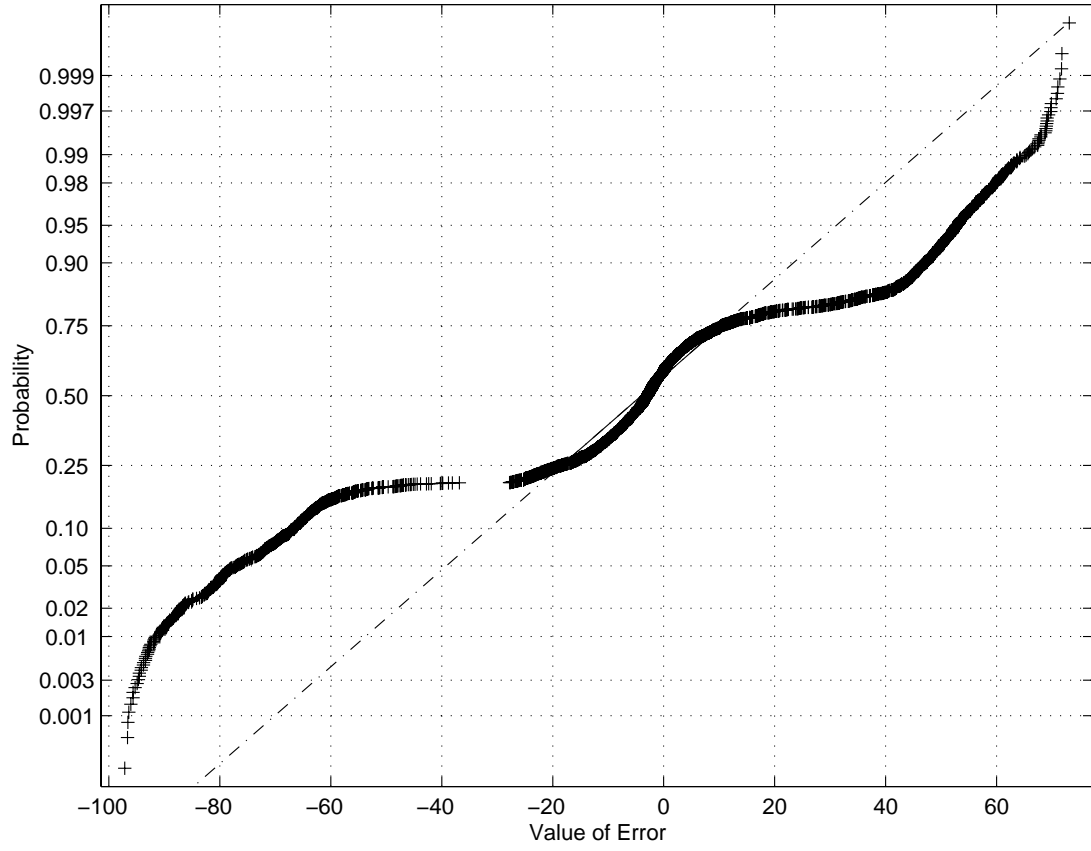


Figure 3.25: Normal probability plot for the errors of \hat{Y}_1 of the VW model.

Table 3.9: Significance Levels of Different Tests on the VW Model.

Tested Models	Normality Tests		Randomization Tests	
	t	F	t	F
\hat{Y}_1, \hat{Y}_2 on $\{s_{17}, \dots, s_{24}\}$	1.2609	2.0781	1.7578	0.7343
\hat{Y}_2, \hat{Y}_3 on $\{s_{25}, \dots, s_{32}\}$	17.3823	24.1834	15.4297	23.2129

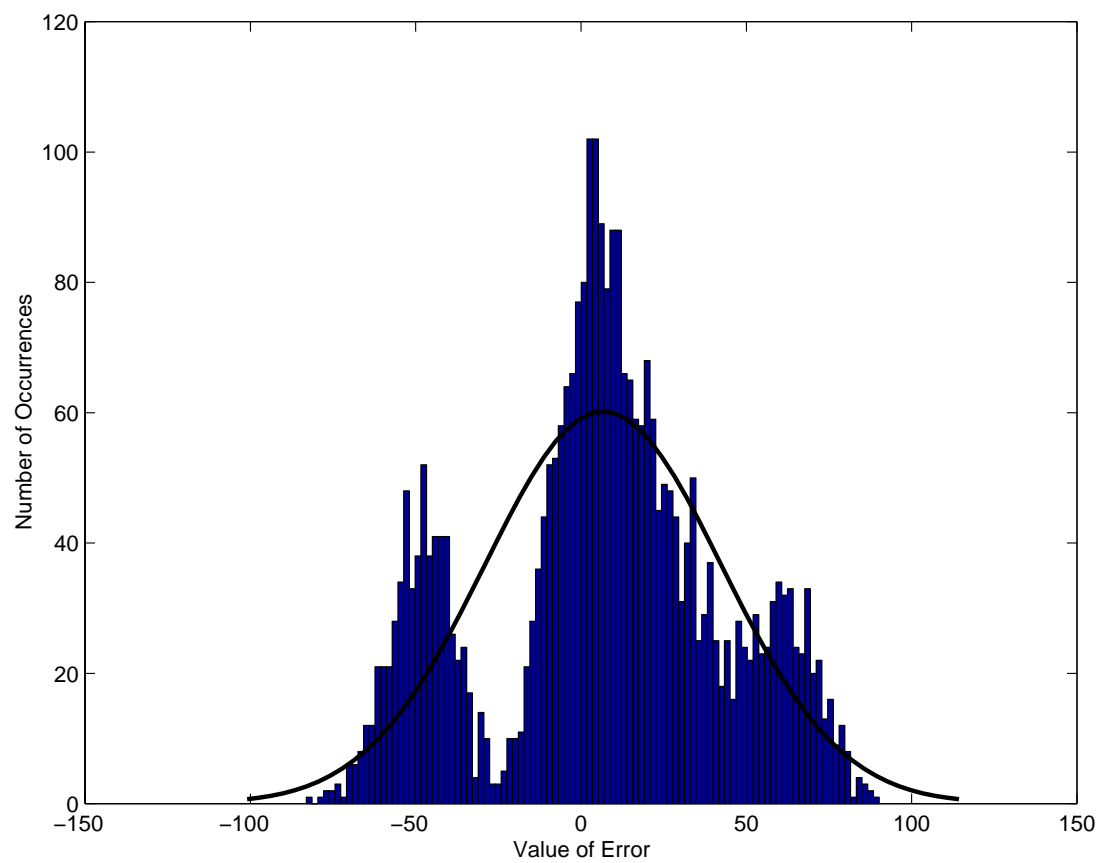


Figure 3.26: Histogram with fitted Normal density for the errors of \hat{Y}_2 of the VW model.

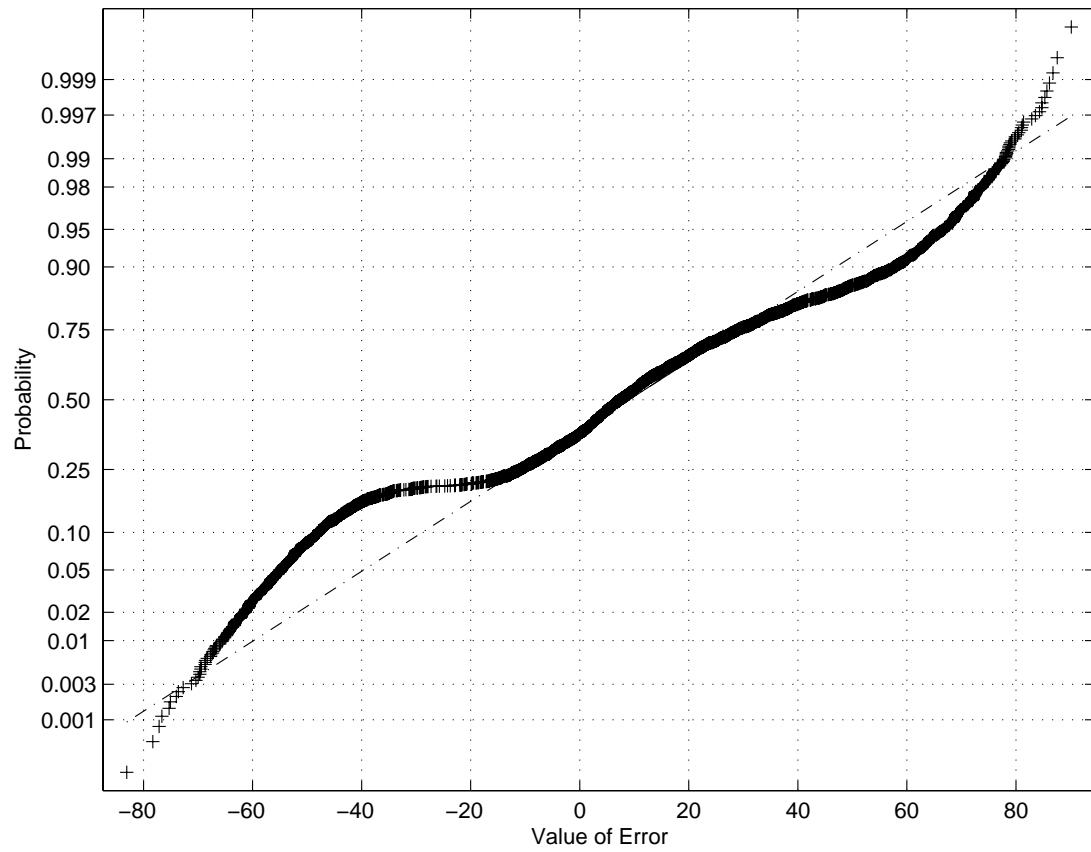


Figure 3.27: Normal probability plot for the errors of \hat{Y}_2 of the VW model.

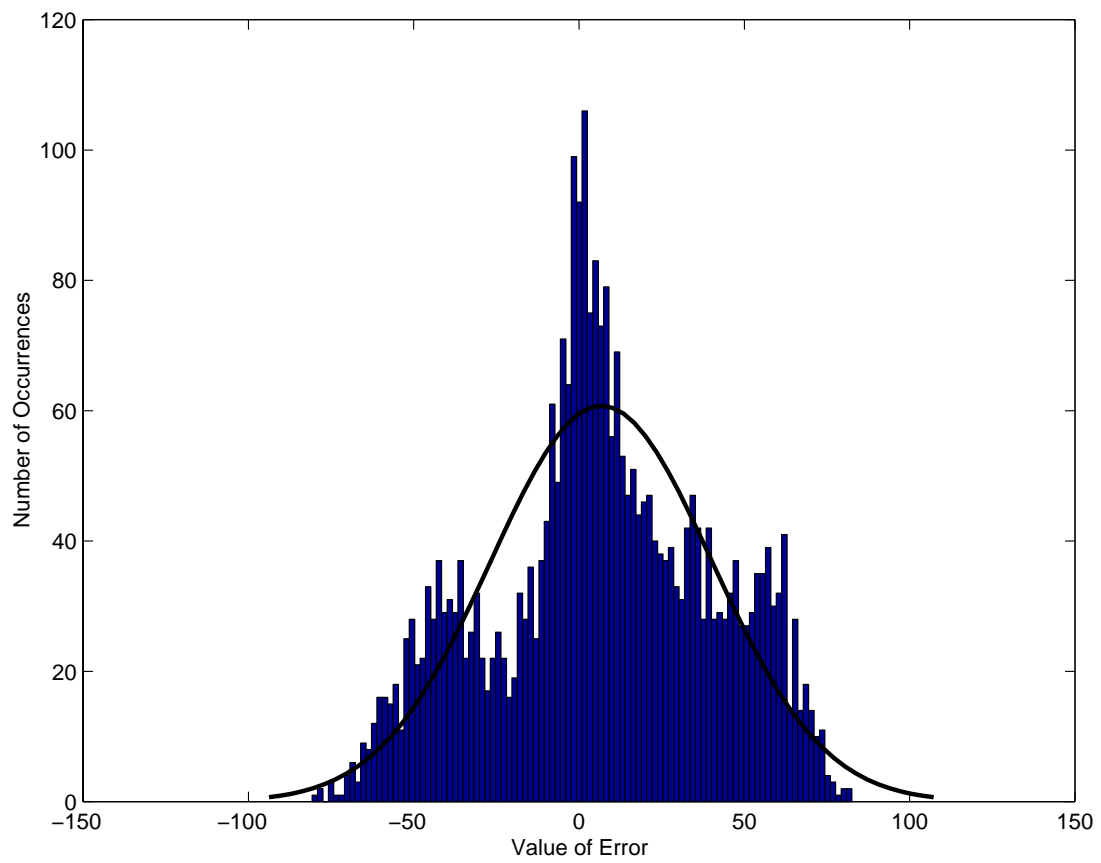


Figure 3.28: Histogram with fitted Normal density for the errors of \hat{Y}_3 of the VW model.

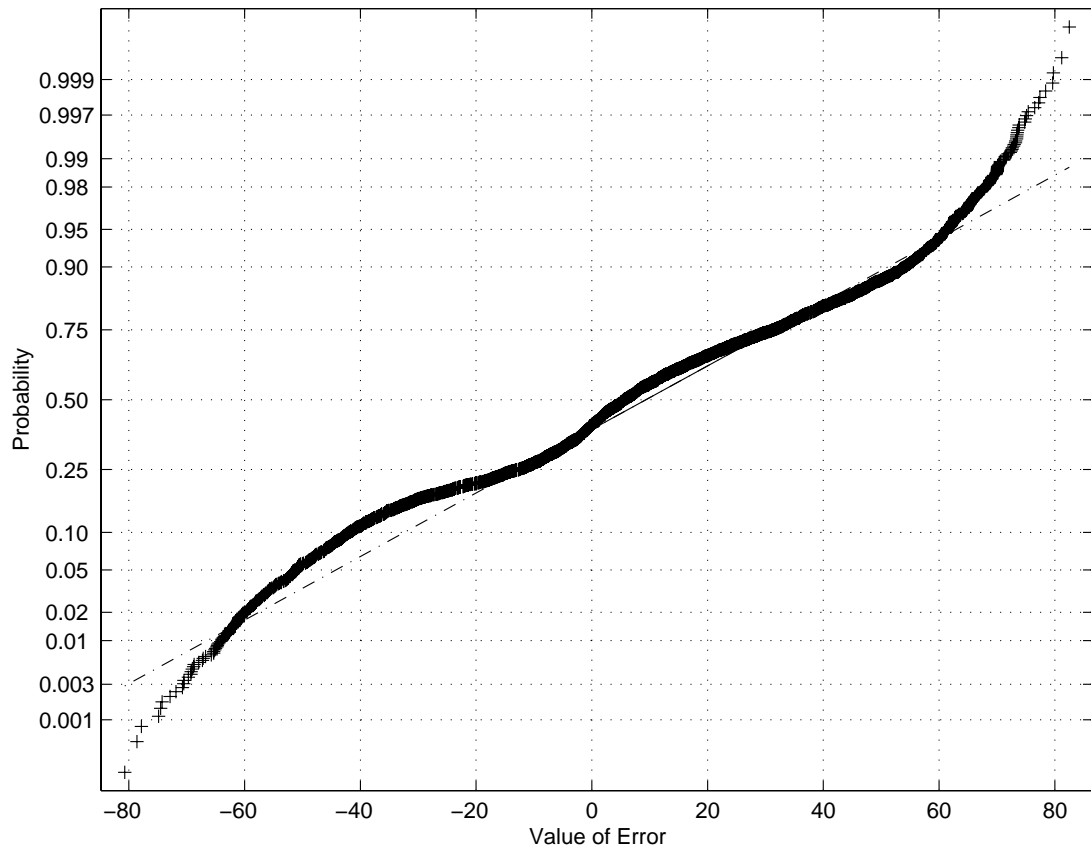


Figure 3.29: Normal probability plot for the errors of \hat{Y}_3 of the VW model.

and the prediction mostly relies on the stochastic error model. But from a viewpoint based on the “main effects principle,” which emphasizes the importance of the linear main factors based on empirical observations, it is reasonable to use the polynomial model containing all main factors as the approximation.

The next step should be experimental design. Among many experimental design methods, the most attractive two are the Latin Hypercube design and the Resolution III fractional factorial design. The advantage of the Latin Hypercube design is that this type of design uses a “space-filling” strategy which is good to study the overall performance of the response, but its disadvantage is that it requires a few more design points and it is not the best design for the linear polynomial model. Alternatively, a Resolution III design is the best design for the polynomial model and it only needs a relatively small number of design points, but it may be outperformed by the Latin Hypercube design because all the design points are corner points. Because multistage metamodels are usually used, the lack of “space-filling” in the Resolution III fractional factorial design can be remedied by adding noncorner design points at later stages. So a Resolution III fractional factorial design is the final choice of the design of experiments for the first metamodel.

Because now there is some knowledge about the underlying response function, an efficient sampling criterion to take advantage of the *a priori* information is expected. Several sampling criteria are tested on a practical finite element model and a randomly generated analytical test function.

From the resulting errors of both test functions in Section 3.8 and 3.9, it can be seen that there is no significant difference between these sampling criteria. The main reason is because the number of design points is small with respect to the dimension of the DVS. If the dimension of the DVS is 1, then the function-fitting problem is almost the same as the nonparametric procedure to estimate a probability density function. The interpolation model used here is similar to the Parzen Windows method in estimating the probability density function without assumption about the form of the density function [15]. With the Parzen Windows method, the estimate is good enough when there are about 20 samples. For a five-dimensional or ten-dimensional DVS, 32 or 64 samples is far fewer than equivalent 20 samples in a one-dimensional DVS. Because the available information is too little, the efficiency of the sampling criterion is not so important. The second reason is that the optimality for any criterion is always based on some assumptions. The optimal design is the one which minimize or maximize the objective function generated from those assumptions. This means that the chosen design is optimal only when the assumptions are valid. But the reality is that no knowledge exists about the function to be fitted which means that the assumptions needed

for the optimality may or may not be valid. So optimality of the chosen design is questionable. For some criteria, there is a need to search for the minimum in a high-dimensional space. The search of m design points in a n -dimensional DVS is not trivial. Based on the difference in the cost and the nonsignificant difference in the efficiency, the random generated design points are a good choice. To avoid the situation that two or more design points are too close to each other, the constraint of minimum distance between design points should be added. At the same time, because the increase in accuracy in the region near added design points can be inferred from the structure of the interpolation model, the sampling range is more crucial than the sampling criterion for newly added design points.

The last problem is how to compare two metamodels. The test with the Normality assumption is simple, but dependent on the validity of the assumption. The distribution-free randomization test does not depend on the distribution assumption of the error variable, but it is costly. So the metamodels should be compared with the randomization method when the computation cost is not high, otherwise the t and F tests with Normality assumption should be used as approximations.

From the standpoint of performance improvement, it can be seen from the resulting significance levels in Sections 3.8 and 3.9 that although the metamodels do not improve the performance much, the confidence in the improvement is relatively strong even with only 8 or 16 test samples for five-dimensional and ten-dimensional DVS, respectively. Because the test design points are randomly chosen, sometimes even when the Normality assumption is not valid as in the case of the Volkswagen model, the results of t and F tests with the Normality assumption are close to those from the randomization tests. The computation cost of the randomization test increases exponentially with the number of test samples. To run the randomization F test with 16 test samples, the F value must be computed for $\frac{(32)!}{(16)! \cdot (16)!} = 601080390$ possible cases. The C program to compute all the F 's runs about 35 minutes on a Sun Ultra 10. If the number of test samples becomes 22, the running time is about 84 days on the same computer. So the t and F tests should be run with both methods when possible, but the tests with the Normality assumption should only be run when the other one is prohibitive.

Two principles underlie the choices above: the parsimony and the robustness of the metamodel. The principle of parsimony is obvious because it is decided by the nature of the preliminary stage engineering design. Robustness here means that the predictive ability of the metamodel should be insensitive to violations of the basic assumption, and it is the major reason for the choice of the linear approximation in the interpolation model and the sampling criterion. For the interpolation

model, even if the probability model of the approximation error is invalid, the approximation still can maintain enough predictive ability because of the “main effects principle.” For the sampling criterion, the optimality of the set of design points does not contribute much to the performance of the metamodel when the number of design points is small with respect to the dimensionality of the DVS. There may also be some negative effects on the performance of the metamodel if the assumption for the optimality is violated. So the random method with constraint on the minimum distance between design points is the choice for the sampling criterion.

The procedure for building a multistage metamodel for one performance variable is as follows:

1. Specify n design variables and their ranges.
2. Construct a Resolution III fractional factorial design S_0 with $\text{card}(S_0) = 2^k \geq n$.
3. Run the simulation software at design points in S_0 .
4. Build the first metamodel using Equation 3.7 with results from last step.
5. Set $i = 1$ and repeat until satisfactory results are achieved:
 - (a) Decide the number of design points to be added at i^{th} stage, $\text{card}(S_i)$.
 - (b) Randomly generate design points S_i with constraint on the minimum distance.
 - (c) Run the simulation software at design points in S_i .
 - (d) Test the improvement of the $(i - 1)^{\text{th}}$ metamodel from the $(i - 2)^{\text{th}}$ metamodel.
 - (e) Build the i^{th} metamodel using Equation 3.7 with results from $S_0 \cup S_1 \cup \dots \cup S_i$.
 - (f) $i = i + 1$.

Chapter 4

Computation of Preference in DVS and PVS

Chapter 2 introduced some basic concepts of imprecision in engineering design problems, discussed the details of the modeling of imprecision with preference, aggregation functions for the preferences, and described how to compute the overall preference. The design preferences, $\mu_{d_i}(d_i)$'s, are specified in DVS and can be aggregated into the combined design preference $\mu_d(\vec{d})$. In similar way, the functional requirements, $\mu_{p_i}(p_i)$'s, are specified in PVS and can be aggregated into the combined functional requirement $\mu_p(\vec{p})$. The combined design preferences and the combined functional requirements are specified in DVS and PVS, respectively. They have to be in the same space in order to aggregate them into the overall preference. Usually the mapping from DVS to PVS, $\vec{p} = \vec{f}(\vec{d})$, is available but computationally expensive. Chapter 3 discussed how to efficiently build a multistage metamodel for the mapping between DVS and PVS. So $\vec{f}(\vec{d})$ can be replaced by its metamodel $\vec{f}'(\vec{d})$ to reduce the computational cost. The overall preference for a design $\mu_o(\vec{p})$ will be determined if $\mu_d(\vec{p})$ can be obtained by mapping $\mu_d(\vec{d})$ onto PVS. Additionally, \vec{f}^{-1} is generally not available, so it is not possible to map the overall preference for a design back onto the DVS from the PVS.

This chapter first introduces the principle of mapping preference, and an implementation of this principle, the *Level Interval Algorithm*, in Section 4.1. Then some anomalies and limitation of the original LIA implementation are discussed in Section 4.2. Section 4.3 introduces some extensions of the original LIA. The methods to compute overall preferences in both DVS and PVS without \vec{f}^{-1} are discussed in Section 4.4.

4.1 The Extension Principle and Level Interval Algorithm (or Vertex Method)

The combined design preference in DVS, $\mu_d(\vec{d})$, and combined functional requirement, $\mu_p(\vec{p})$, can be obtained by aggregation in the corresponding spaces. Finally $\mu_d(\vec{d})$ and $\mu_p(\vec{p})$ will use a trade-off aggregation to get μ_o , the overall preference. But $\mu_d(\vec{d})$ and $\mu_p(\vec{p})$ are expressed in different spaces. One of the combined preferences need to be mapped into the other space before computing μ_o . The mapping functions from DVS to PVS, $\vec{p} = \vec{f}(\vec{d})$, are usually available, so here $\mu_d(\vec{p})$, the combined design preference in PVS, is considered to be induced from $\mu_d(\vec{d})$ using the extension principle [66]. If $\vec{f}^{-1}(\vec{p})$ is available, the preference in DVS can also be computed from the preference in PVS by using the extension principle.

$$\mu_d(\vec{p}) = \sup\{ \mu_d(\vec{d}) \mid \vec{p} = \vec{f}(\vec{d}) \} \quad (4.1)$$

where sup over the null set is defined as zero.

There are many ways to implement the extension principle. One way is to solve this problem analytically and exactly. Baas and Kwakernaak [2] consider it as a nonlinear programming problem [2]. Consider one single performance variable $p_j = f_j(\vec{d})$. The problem is to maximize $\mu_d(p_j)$ with the following constraints:

$$\begin{cases} \mu_d(p) \leq \mu_d(d_i), & i = 1 \dots n \\ p_j = f_j(\vec{d}) \end{cases} \quad (4.2)$$

If some conditions are met [2], this problem can be solved. But in general it is difficult to solve this nonlinear programming problem for arbitrary $f_j(\vec{d})$. There are also some approximate analytical methods to implement the extension principle. One method proposed by Dubois and Prade [13] simplifies the equation by dividing the membership functions into the left side and right side. This approximate method provides good results, but the accuracy decreases when extended division is needed. Schumucker [47] proposed an approximate numerical method, which discretizes the supports of design preferences and uses the preferences at these locations to compute $\mu_d(p_j)$. Because of the nonlinear nature of the sup operation, the result from the approximate numerical method is not always good enough, and even the revised version with imposed convexity does not work well.

There is another approximate numerical method called the *Level Interval Algorithm*, or *LIA*, and sometimes it is also called the *Vertex Method*. It was first proposed by Dong and Wong [12] in order to solve the extended operation in the weighted average operation. This method also makes use of discretization, but it works on the membership value instead of the support. It also uses interval analysis to get the final solution. The LIA algorithm is summarized below.

First, a set of M discretized preference values, $\alpha_k, k = 1..M$, are specified. For each α_k , an interval, $[d_{i,\min}^{\alpha_k}, d_{i,\max}^{\alpha_k}], i = 1 .. n$, will be generated for each design preference function. The α -cut $D_{\alpha_k}^d$ of the combined design preference are defined as

$$D_{\alpha_k}^d = \{\vec{d} \in DVS \mid \mu_d(\vec{d}) \geq \alpha_k\} \quad k = 1..M. \quad (4.3)$$

It is assumed to be the Cartesian product of the n intervals for individual design preference functions.

$$D_{\alpha_k}^d = [d_{1,\min}^{\alpha_k}, d_{1,\max}^{\alpha_k}] \times \cdots \times [d_{n,\min}^{\alpha_k}, d_{n,\max}^{\alpha_k}] \quad (4.4)$$

There are 2^n corner points of $D_{\alpha_k}^d$, which are permutations of the end points of the individual design preference intervals. All these corner points are mapped onto PVS with $p_j = f_j(\vec{d})$, and 2^n values of p_j are obtained. Find the minimum value $p_{j_{\min}}^{\alpha_k}$ and the maximum value $p_{j_{\max}}^{\alpha_k}$ among them, then the α -cut of p is the interval:

$$[p_{j_{\min}}^{\alpha_k}, p_{j_{\max}}^{\alpha_k}] = \{f_j(\vec{d}) \mid \mu_d(\vec{d}) \geq \alpha_k\} \quad (4.5)$$

A simple example is used to illustrate the LIA. For a design problem with two design variables d_1 and d_2 , and one performance variable $p = f(\vec{d}) = (d_1 + 2)^3 - 6 \cdot d_2$. The preferences for both design variables are the same as shown in Figure 4.1 with three α levels, $\{\varepsilon, 0.5, 1.0\}$.

For each α level, 4 corner points in the DVS are mapped onto the PVS, and the α -cut of $\mu_d(p)$ is specified by the minimum performance and the maximum performance. The resulting $\mu_d(p)$ represented by three α -cuts is shown in Figure 4.2.

4.2 Limitation of original LIA for the Mapping between DVS and PVS

Although the LIA is an effective implementation of the extension principle, its good performance is based on some assumptions. If these assumptions are violated, the LIA will generate poor results

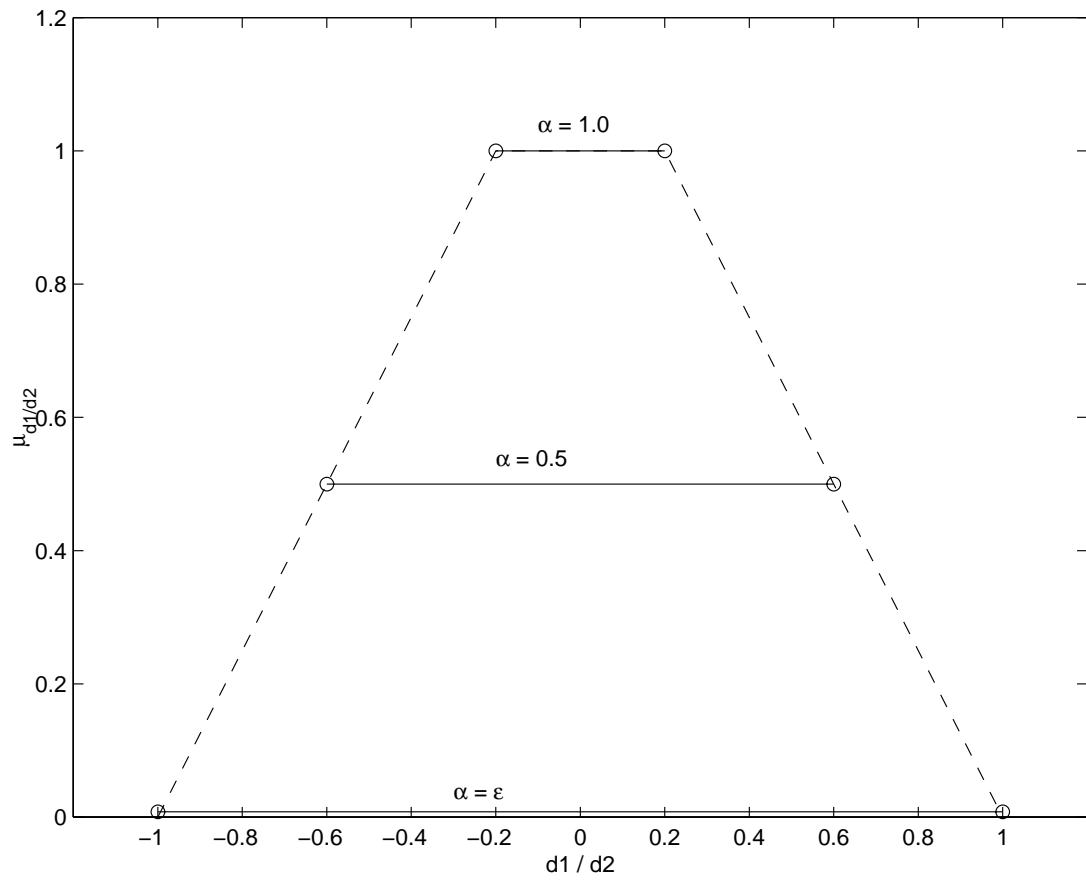


Figure 4.1: The preference function of the design variable d_1 or d_2 .

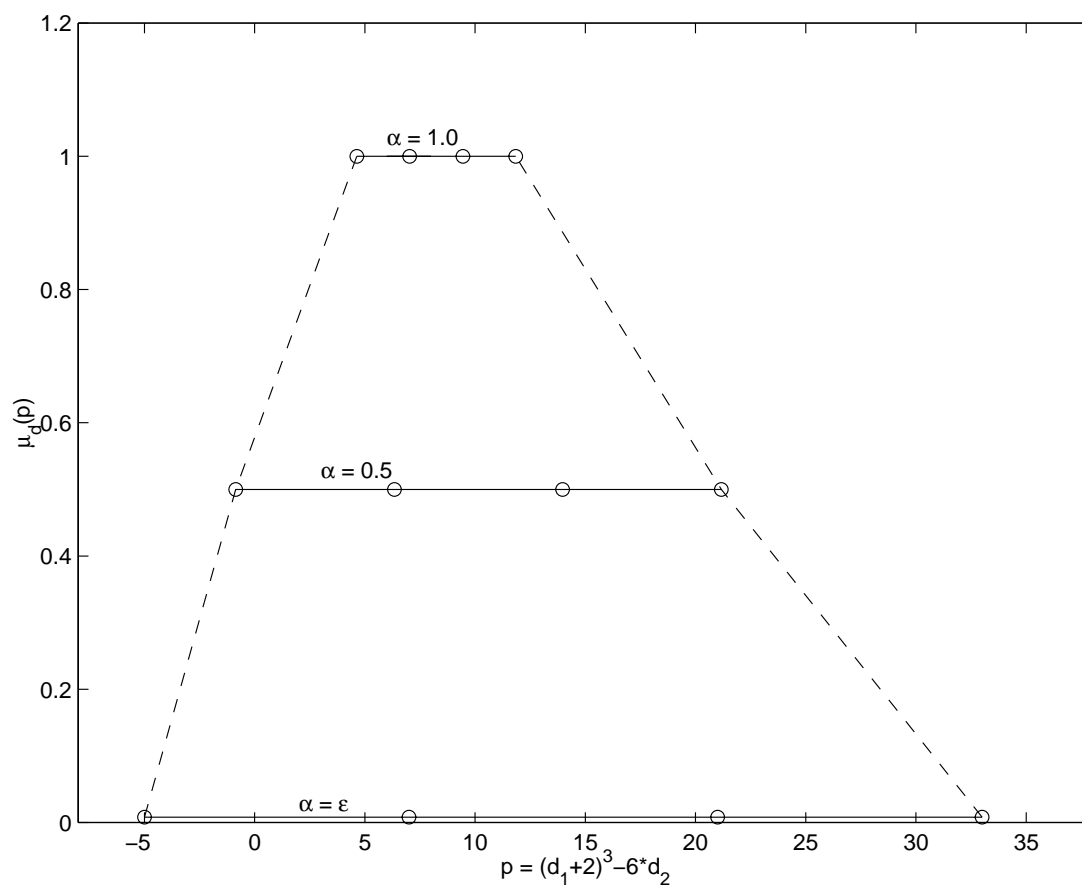


Figure 4.2: Example result of LIA.

or even wrong results. This section will discuss the limitations of the LIA.

4.2.1 Anomalies in the LIA for a Single Preference Function

To simplify the situation, consider the limitations of the LIA in the situation where it is desired to compute one performance variable induced from one design variable, *i.e.*, map from $\mu_d(d)$ to $\mu_d(p)$ with $p = f(d)$. First, assume that the design preference functions have normality and convexity, and are continuous over the interested region of d . For such well-defined $\mu_d(d)$, there are several possible anomalies if the LIA is used to compute $\mu_d(p = f(d))$ [64].

1. $\mu_d(f(d))$ will be infinite if $f(d)$ is unbounded within the support of $\mu_d(d)$. For example, $f(d) = 1/d$.
2. $\mu_d(f(d))$ will be difficult to compute if $f(d)$ has singularity within the support of $\mu_d(d)$. For example, $f(d) = \sin(1/d)$. Details can be found in [64].
3. $\mu_d(f(d))$ will be uninterpretable, *i.e.*, will oscillate near some limit values, if $f(d)$ is infinitely multivalued within the support of $\mu_d(d)$.
4. $\mu_d(f(d))$ will have nonsensible results if $f(d)$ has finite extrema within the support of $\mu_d(d)$.

In the M_QI, the metamodel of $\vec{f}(\vec{d})$, $\vec{f}'(\vec{d})$, instead of $\vec{f}(\vec{d})$ is used. Because of the model structure and base functions of $\vec{f}'(\vec{d})$, $\vec{f}'(\vec{d})$ is bounded and finite multivalued, and will not have singularities within the support of $\mu_d(d)$. So the only possible anomaly will arise when $\vec{f}'(\vec{d})$ is nonmonotonic within the support of $\mu_d(\vec{d})$.

Consider the following simple triangle preference function $\mu_d(d)$ shown in Figure 4.1. The performance function is a simple cubic polynomial as following:

$$p = f(d) = 3.0 \cdot d^3 - 2.5 \cdot d$$

and is shown in Figure 4.3.

Using the LIA described in Section 4.1, α -levels $\varepsilon, 0.1, 1$ are used to compute the preference on the performance variable, $\mu_d^d(p)$. The result is shown in Figure 4.4.

From the result it can be seen that the α -cuts of $\mu_d(p)$ with $\alpha \leq 0.5$ are not accurate. This is because the performance function $f(d)$ has local extrema for $-1 \leq d \leq 1$. For α -cuts $\alpha_k = k/10$, $0 \leq k \leq 10$, $D_{\alpha_k}^d = [d_{\min}^{\alpha_k}, d_{\max}^{\alpha_k}] = [-(1 - \frac{4}{5}\alpha_k), 1 - \frac{4}{5}\alpha_k]$. The length of $P_{\alpha_k}^d$ is $|f(d_{\max}^{\alpha_k}) -$

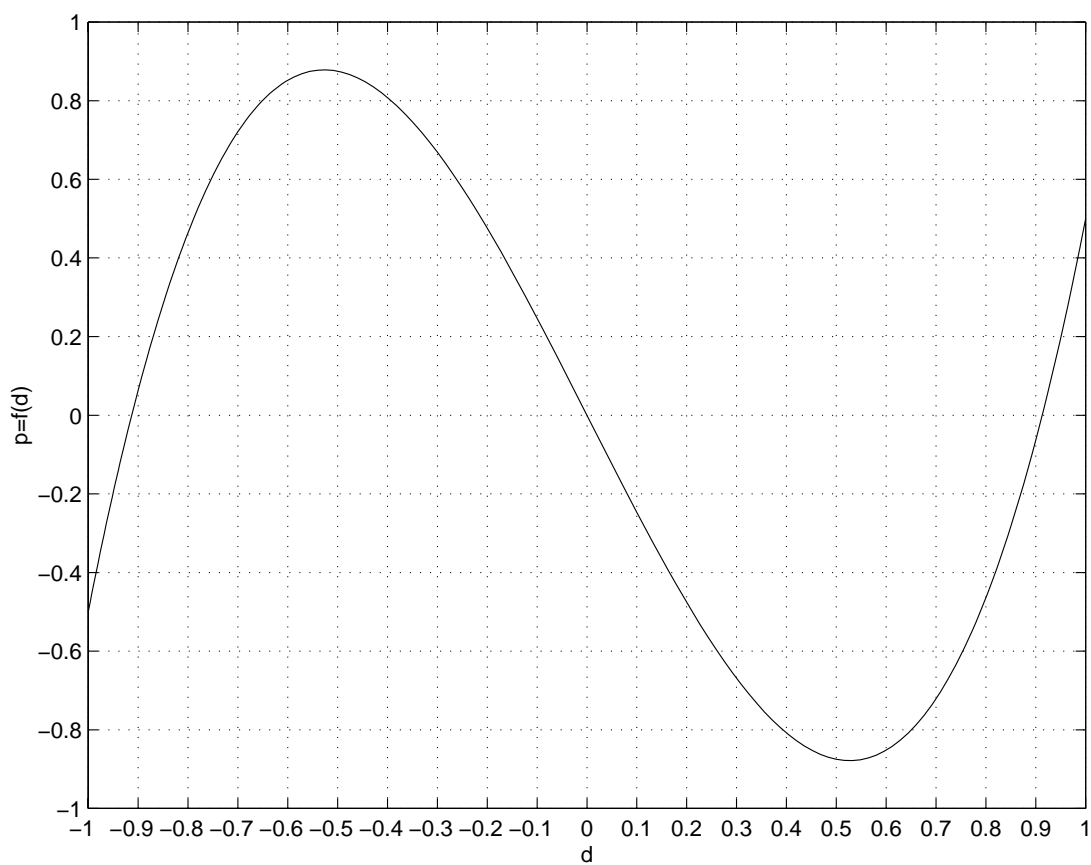


Figure 4.3: The performance function $p = f(d)$.

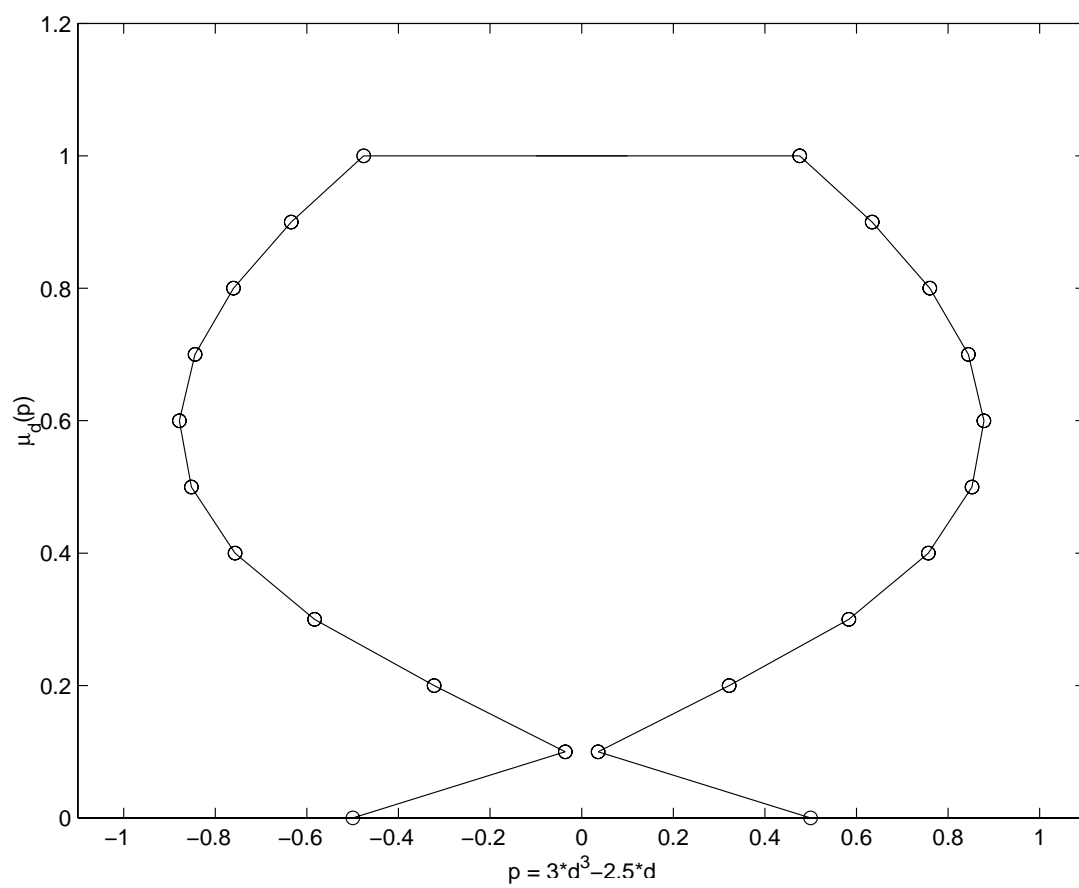


Figure 4.4: $\mu_d(p)$ for μ_d as in Figure 4.1, where $p = 3d^3 + 2.5d$.

$|f(d_{\min}^{\alpha_k})| = |f(1 - \frac{4}{5}\alpha_k) - f(-(1 - \frac{4}{5}\alpha_k))| = 2 \cdot |f(1 - \frac{4}{5}\alpha_k)|$ because $f(d)$ is an odd function of d . For $0 \leq k \leq 10$, $|f(1 - \frac{4}{5}\alpha_k)| = |f(1 - \frac{4}{5}k/10)|$ will reach the maxima at $k = 6$. So for $1 \leq k \leq 5$, $P_{\alpha_6}^d \supset P_{\alpha_k}^d$, but $\alpha_6 < \alpha_k$. Thus, it can be seen that the internal extrema of $f(d)$ causes this anomaly in the results of the LIA.

4.2.2 Limitations of LIA for Multiple Design Preferences

For a multidimensional DVS, even if every single design preference, $\mu_{d_i}(d_i)$, does not cause any of the anomalies listed in Section 4.2.1, there still may be some errors when the LIA is applied to the combined design preference.

When using the LIA for a multidimensional DVS, D_{α_k} is assumed to be the Cartesian product of the individual intervals of design variables. This is only accurate under some special cases, such as $\mathcal{P} = \min$, $\alpha_k = \{\varepsilon, 1.0\}$, or $\mathcal{P} \neq \max$ and $\alpha_k = 1.0$. Otherwise, D_{α} is not a hypercube as assumed.

Consider a two-dimensional DVS. The preference functions on two design variables, d_1 and d_2 , are the same, as shown in Figure 4.1 with α -cuts at $\{\varepsilon, 0.5, 1.0\}$. Then the $\mu(d_1, d_2)$'s are computed by several aggregation functions: $\min(\mu_{d1}, \mu_{d2})$, $\max(\mu_{d1}, \mu_{d2})$, $(\mu_{d1} + \mu_{d2})/2$, and $\sqrt{\mu_{d1} \cdot \mu_{d2}}$. All four aggregated preference functions are shown in Figure 4.5 with D_{α_k} at $\alpha_k = \{\varepsilon, 0.5, 1.0\}$. The D_{ε} 's are always rectangles, as assumed no matter which aggregation function is used because of the annihilation property. All D_{α_k} 's from $\mathcal{P} = \min$ are all rectangles as assumed. In the results of $\mathcal{P} = (\mu_{d1} + \mu_{d2})/2$ and $\mathcal{P} = \sqrt{\mu_{d1} \cdot \mu_{d2}}$, $D_{1.0}$ are also rectangles. But $D_{1.0}$ is not a rectangle in the result of $\mathcal{P} = \max(\mu_{d1} + \mu_{d2})$. And D_{α_k} 's with $\alpha_k = 0.5$ or any other $0 < \alpha_k < 1.0$ from aggregation functions other than $\mathcal{P} = \min(d_1, d_2)$ are not rectangles. It can also be noticed that the assumed rectangle will be smaller than the actual D_{α_k} -cut if the assumption is violated.

To avoid this limitation of the LIA, the values of α can be limited. to ε and 1.0. However, sometimes the intermediate α -levels are also necessary to find the relevant set of designs. Also, sometimes the preference function does not reach 1.0, especially for the overall preference function μ_o .

4.2.3 Limitations of the LIA for Multiple Performance Variables

By using the LIA, the α -cut for the preference on any single performance variable can be computed. If there are $q > 1$ performance variables, for any α -level there will be q α -cuts for the performance variables, $\{ [p_{1,\min} \dots p_{1,\max}], \dots, [p_{q,\min} \dots p_{q,\max}] \}$. One simple way to generate $P_{\alpha_k}^d$ is to use the

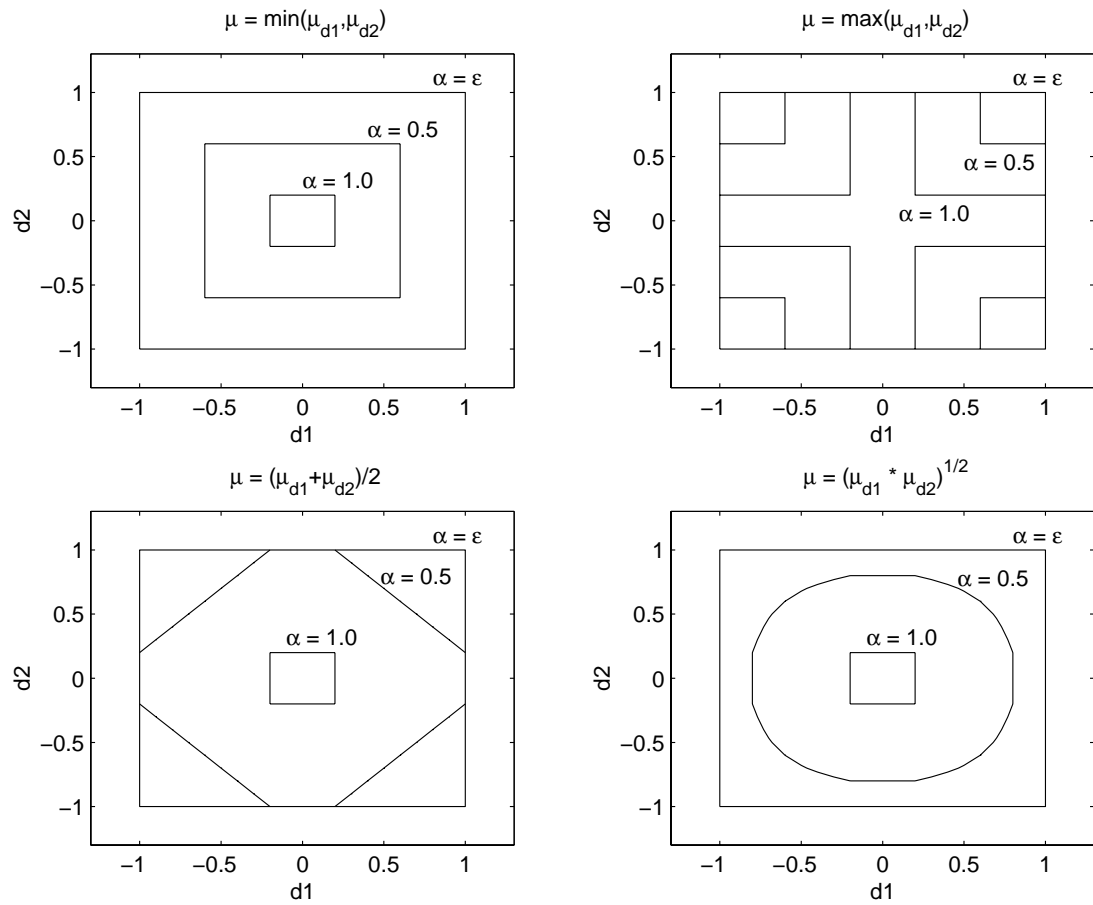


Figure 4.5: The combined design preferences of two design variables by different \mathcal{P} s.

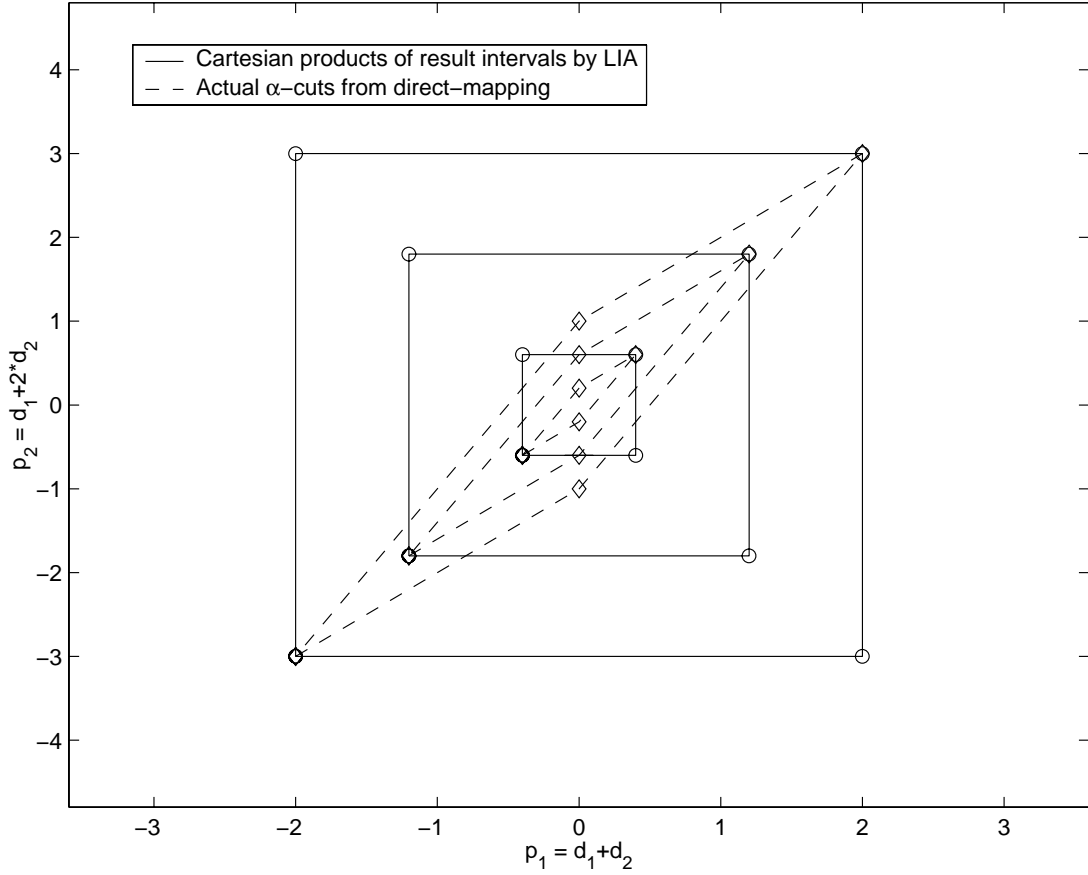


Figure 4.6: $P_{\alpha_k}^d$'s in a 2-D PVS from a 2-D DVS.

Cartesian product of the q intervals:

$$P_{\alpha_k}^d = [p_{1,\min} \cdots p_{1,\max}] \times \cdots \times [p_{q,\min} \cdots p_{q,\max}] \quad (4.6)$$

Even if the mappings of the design variables to the performance variables do not cause any anomalies of the LIA, and the aggregation function for the design variables does not create α -cuts different from hypercubes, the result from Equation 4.6 is accurate only when there is no dependence between any two mapping functions from the DVS to the PV's. Otherwise there are some distortions from the actual α -cut.

For example, consider a simple system of two design variables, d_1 and d_2 , and two performance variables, p_1 and p_2 . The design preferences are the same as those in Figure 4.1. The aggregation function is $\mathcal{P} = \min(\mu_{d1}, \mu_{d2})$, so the resulting α -cuts are exactly the same as those in the upper-left figure in Figure 4.5. The mappings from the DVS to the PVS are:

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} f_1(d_1, d_2) = d_1 + d_2 \\ f_2(d_1, d_2) = d_1 + 2 \cdot d_2 \end{pmatrix} \quad (4.7)$$

Within the region $[-1 \dots 1] \times [-1 \dots 1]$, the mapping functions, f_1 and f_2 , are single-valued and bounded, have no singularities, and are also monotonic. So the results of $[p_{1,\min} \dots p_{1,\max}]$ and $[p_{2,\min} \dots p_{2,\max}]$ using the LIA are accurate. The result is that $P_{\alpha_k}^d$ is a Cartesian product of the intervals on p_1 and p_2 .

Because f_1 and f_2 are both linear functions of d_1 and d_2 , an alternate way to compute $P_{\alpha_k}^d$ is to map the corner points of $D_{\alpha_k}^d$ to the PVS and connect them in the same order as in the DVS. The two sets of $P_{\alpha_k}^d$'s for $\alpha = \{\varepsilon, 0.5, 1.0\}$ are shown in Figure 4.6. There are significant differences between the α -cuts generated by the different methods because of the strong dependence between f_1 and f_2 . The stronger the dependence between mapping functions, the bigger the difference between the results from the two methods. The situation will become more complicated if the mapping functions are not linear, or there are other errors caused by the anomalies of the LIA or the aggregation function in DVS.

4.3 The Revised LIA

There are many possible anomalies and limitations of the LIA described in Section 4.2. There are also many extensions of the LIA to improve its performance.

Among all four types of anomalies listed in section 4.2.1, the first three can be avoided by using the metamodel $\vec{f}'(\vec{d})$ of $\vec{f}(\vec{d})$, because its model structure and the chosen base functions ensured that no anomalies will happen except the fourth one ($f(d)$ has finite extrema).

Wood, Otto, and Antonsson [64] proposed an extension of the LIA to remedy the anomaly caused by finite number of internal extrema of $f_j(\vec{d})$ within the support of any d_i . This extended method tries to find the extrema of the mapping function within or on the boundary of the α -cut of \vec{d} . Then these extrema are used to improve the $P_{\alpha_k}^d$.

When the dimensionality of the DVS is high, it is expensive to find the extrema by solving the equation $\frac{\partial p_j}{\partial d_i} = 0$ directly. Law and Antonsson [30, 31] proposed a simplified method to find the extrema. First build a linear approximation for all f_j 's. Then evaluate f_j at the center point of the DVS (\vec{d}_{ctr}), and use $f_j(\vec{d}_{ctr})$ to detect nonlinear f_j 's. Finally refine the linear approximation for all the linear f_j 's. Then the extrema can be found by using conventional optimization method to find

the extrema in the reduce search space of only those nonlinear f_j 's.

The linear approximation of all f_j 's can also be used to relax the assumption that $P_{\alpha+k}^d$ is a hypercube. The linear approximation is in the form as [31]:

$$\begin{aligned}\vec{f}'(\vec{d}) &= A \cdot (\vec{d} - \vec{d}_{ctr}) + \vec{\Delta} \\ &= \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{q1} & \cdots & a_{qn} \end{bmatrix} \cdot \begin{bmatrix} d_1 - d_1^{ctr} \\ \vdots \\ d_n - d_n^{ctr} \end{bmatrix} + \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_n \end{bmatrix}\end{aligned}\quad (4.8)$$

A polyhedron can be generated to approximate the actual $P_{\alpha k}^d$. Consider the column vectors in matrix A as the principle directions, and the extrema points found through linear approximation or optimization as the corner points of the polyhedron. The result of the extension of the LIA by principle directions of $\vec{f}'_{linear}(\vec{d})$ is denoted as $P_{\alpha k}^{d\sharp}$. For the example in section 4.2.3, the upper-right corner points are the maxima for p_1 and p_2 , the lower-left corner points are the maxima for p_1 and p_2 because both performance functions are linear. Therefore, the actual α -cuts will be found. However, if any extremum is generated by optimization, a special adjustment is needed. The resulting α -cut by the extension of the LIA with linear approximations and optimizations on nonlinear performance functions is denoted as $P_{\alpha k}^{d\sharp}$.

If the f_2 in Equation 4.7 is changed to make a new $f_2^*(d_1, d_2)$ as in Equation 4.9, then the linear approximation of f_2^* is the f_2 in Equation 4.7, and the $P_{\epsilon}^{d\sharp}$ is the same as that in Figure 4.6. If optimization is applied on nonlinear performance functions, the extrema of p_2 will be found as $p_{2,min} = f_2^*(-1.0, -0.75) = -3.5$ and $p_{2,max} = f_2^*(1.0, 0.5) = 3.5$. The extrema of p_1 are the upper-right and the lower-left corner points in Figure 4.6. Two P_{ϵ}^d 's from two different extensions are shown in Figure 4.7. The optimizations add two more regions to $P_{\epsilon}^{d\sharp}$, and some adjustments are needed if convexity is desired. From the location of $(f_1(0.8, 0.8), f_2^*(0.8, 0.8))$, it can be seen that the $P_{\epsilon}^{d\sharp}$ is more accurate than $P_{\epsilon}^{d\sharp}$, because $(0.8, 0.8) \in D_{\epsilon}^d$.

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} f_1(d_1, d_2) = d_1 + d_2 \\ f_2^*(d_1, d_2) = \begin{cases} d_1 - d_2 + 3.0 & \text{if } d_2 \geq 0.5 \\ d_1 + 4d_2 + 0.5 & \text{if } -0.75 \geq d_2 > -0.5 \\ d_1 - 2d_2 - 4.0 & \text{if } -0.75 > d_2 \end{cases} \end{pmatrix} \quad (4.9)$$

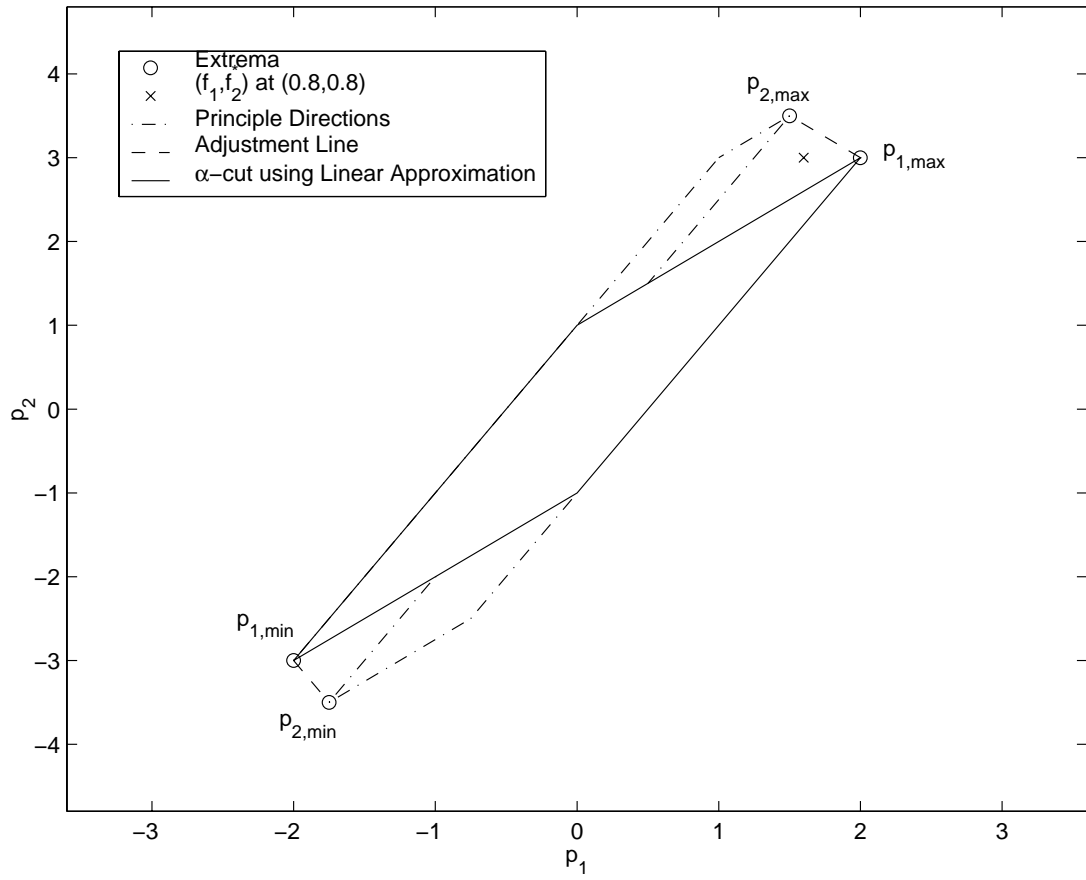


Figure 4.7: $P_\varepsilon^{d_H}$ and $P_\varepsilon^{d_H}$ in a 2-D PVS from a 2-D DVS.

For a mapping between a two-dimensional DVS and a two-dimensional PVS, the result generated by the extrema and principle directions with adjustment lines is highly accurate. However for a mapping between an n -dimensional DVS and a q -dimensional PVS, if $n > q$, then $P_{\alpha_k}^d$ has q degrees of freedom and its boundary has $q - 1$ degrees of freedom. For $q \geq 3$, it is difficult to generate the boundary of $P_{\alpha_k}^d$ from the principle directions which have only 2 degrees of freedom, and it is also difficult to add the adjustment patches for the resulting extrema from the optimizations.

Although the extension of the LIA by a linear approximation of $\vec{f}(\vec{d})$ can improve the performance of the LIA, the resulting α -cut is not presented in a desired form. The operations on q -dimensional polyhedron are not trivial even if the difficulty of constructing the polyhedra is ignored, and patches are added, when some extrema are the results of optimizations. On the other hand, the results from Equation 4.4 and Equation 4.6 are less accurate but expressed in a much simpler form, n -cubes. Operations on n -cubes are much easier than those on n -dimensional polyhedra. Furthermore, the errors caused by invalid assumptions about the aggregation function or the independence between f_j 's and the chance that f_j 's have local extrema within DVS are all proportional to the size or volume of the DVS or PVS. Based on above the observations, an alternative way to improve the performance of the LIA by dividing the DVS and PVS into smaller regions is proposed below.

First, divide the relevant range of design variable d_i into s_i subregions by $\{d_{i,0}^\diamond, \dots, d_{i,s_i}^\diamond\}$, The subregion for d_i , the sub-hypercube in the DVS and its center points, are denoted by

$$\begin{aligned} \mathcal{X}_{i,r_i} &= [d_{i,r_i-1}^\diamond, d_{i,r_i}^\diamond] \\ \mathcal{X}_{\vec{r}}^* &= \mathcal{X}_{1,r_1} \times \dots \times \mathcal{X}_{n,r_n} \\ \vec{c}_{\vec{r}} &= (c_{\vec{r},1} \dots c_{\vec{r},n}) \\ &= \left((d_{1,r_1}^\diamond - d_{1,r_1-1}^\diamond)/2, \dots, (d_{n,r_n}^\diamond - d_{n,r_n-1}^\diamond)/2 \right) \\ \text{where } \vec{r} &= (r_1, \dots, r_n), \quad 1 \leq r_i \leq s_i, \quad 1 \leq i \leq n \end{aligned} \tag{4.10}$$

Now each sub-hypercube will have its local design preference, which includes the effect of the aggregation function:

$$\mu_{d_i, \vec{r}}(d_i) = \mathcal{P}(\mu_{d_1}(c_{\vec{r},1}), \dots, \mu_{d_i}(d_i), \dots, \mu_{d_n}(c_{\vec{r},n})) \tag{4.11}$$

where $\vec{r} = (r_1, \dots, r_n)$, $1 \leq r_i \leq s_i$, $1 \leq i \leq n$

Then find the α -cut of the local design preference $\mu_{d_i, \vec{r}}(d_i)$ in each $\mathcal{X}_{\vec{r}}$ as $[d_{i, r_i}^{\min}, d_{i, r_i}^{\max}]$, and use Equation 4.4 compute $D_{\vec{r}, \alpha_k}^d$. The whole α -cut in the DVS, denoted as $D_{\alpha_k}^{d\Box}(\vec{s})$, is the union of all $D_{\vec{r}, \alpha_k}^d$ over all sub-hypercubes in the DVS. If $s_i = \mathcal{S}$, $\forall 1 \leq i \leq n$, the whole α -cut is denoted as $D_{\alpha_k}^{d\Box}(\mathcal{S})$.

$$\begin{aligned} D_{\vec{r}, \alpha_k}^d &= [d_{1, r_1, \min}^{\alpha_k}, d_{1, r_1, \max}^{\alpha_k}] \times \dots \times [d_{n, r_n, \min}^{\alpha_k}, d_{n, r_n, \max}^{\alpha_k}] \\ D_{\alpha_k}^{d\Box}(\vec{s}) &= \bigcup_{\forall \vec{r}} D_{\vec{r}, \alpha_k}^d \\ \text{where } \vec{r} &= (r_1, \dots, r_n), \vec{s} = (s_1, \dots, s_n) \\ \text{and } 1 &\leq r_i \leq s_i, 1 \leq i \leq n \end{aligned} \quad (4.12)$$

The $P_{\vec{r}, \alpha_k}^d$ can be computed in a similar way. First, compute $D_{\alpha_k}^{d\Box}$ as described above. Then divide the relevant range of each design variable p_i into u_i subregions by $\{p_{i,0}^\diamond, \dots, p_{i,u_i}^\diamond\}$. The subregion for p_j and the sub-hypercube formed by these subregions are denoted by

$$\begin{aligned} \mathcal{Y}_{j, t_j} &= [p_{j, t_j - 1}^\diamond, p_{j, t_j}^\diamond] \\ \mathcal{Y}_{\vec{t}}^* &= \mathcal{Y}_{1, t_1} \times \dots \times \mathcal{Y}_{q, t_q} \\ \text{where } \vec{t} &= (t_1, \dots, t_q), 1 \leq t_j \leq u_j, 1 \leq j \leq q \end{aligned} \quad (4.13)$$

Then apply the original LIA to each $D_{\vec{r}, \alpha_k}^d$ to get the α -cut on each p_j in each $\mathcal{Y}_{\vec{t}}^*$ as $[p_{j, t_j, \vec{r}, \min}^{\alpha_k}, p_{j, t_j, \vec{r}, \max}^{\alpha_k}]$. The endpoints of the α -cut of $\mu_d(p_j)$ in $\mathcal{Y}_{\vec{t}}^*$ are the union of the α -cuts in all $\mathcal{X}_{\vec{r}}^*$ s:

$$\begin{aligned} p_{j, t_j, \min}^{\alpha_k} &= \min_{\forall \vec{r}} p_{j, t_j, \vec{r}, \min}^{\alpha_k} \\ p_{j, t_j, \max}^{\alpha_k} &= \max_{\forall \vec{r}} p_{j, t_j, \vec{r}, \max}^{\alpha_k} \\ \text{where } \vec{r} &= (r_1, \dots, r_n), 1 \leq r_i \leq s_i, 1 \leq i \leq n \\ \text{and } 1 &\leq t_j \leq u_j, 1 \leq j \leq q \end{aligned} \quad (4.14)$$

Then compute the α -cut in each sub-hypercube in the PVS $P_{\vec{t}, \alpha_k}^d$ using Equation 4.6. The α -cut in the whole the PVS, $P_{\alpha_k}^{d\Box}(\vec{s}, \vec{u})$, is the union of all $P_{\vec{t}, \alpha_k}^d$ over all sub-hypercubes in PVS. And if $s_i = \mathcal{S}$, $\forall 1 \leq i \leq n$ and $u_j = \mathcal{U}$, $\forall 1 \leq j \leq q$, the α -cut for the whole PVS is denoted as $P_{\alpha_k}^{d\Box}(\mathcal{S}, \mathcal{U})$:

$$\begin{aligned}
 P_{\vec{t}, \alpha_k}^d &= [p_{1, t_1, \min}^{\alpha_k} \cdot p_{1, t_1, \max}^{\alpha_k}] \times \cdots \times [p_{q, t_q, \min}^{\alpha_k} \cdot p_{q, t_q, \max}^{\alpha_k}] \\
 P_{\alpha_k}^{d\Box}(\vec{s}, \vec{u}) &= \bigcup_{\forall \vec{t}} P_{\vec{t}, \alpha_k}^d \\
 \text{where } \vec{t} &= (t_1, \dots, t_q), \vec{u} = (u_1, \dots, u_q) \\
 &1 \leq t_j \leq u_j, 1 \leq j \leq q \\
 \text{and } \vec{s} &= (s_1, \dots, s_n),
 \end{aligned} \tag{4.15}$$

This extension of the LIA will be demonstrated on the examples in Section 4.2.2 and in Section 4.2.3. For the problem of $\mu_d(\vec{d}) = \mathcal{P}(\mu_{d_1}, \mu_{d_2})$, the support of each design variable is divided equally into 10 subregions, and the interval of each design variable is computed from $\mu_{d_i, \vec{r}}(d_i)$ in each sub-rectangle in the DVS. Then the α -cut is generated by Equation 4.13. The $D_{0.5}^{d\Box}(10)$'s for 4 different aggregation functions are shown in Figure 4.8 with the actual α -cuts. For $\mathcal{P} = \min(d_1, d_2)$, the result from the LIA with the hypercube assumption is correct, so is the result from the extended LIA. For $\mathcal{P} = \max(d_1, d_2)$, the result from the extended LIA is the same as the actual α -cut because the boundaries of the actual α -cut are parallel to the boundaries of the sub-rectangle. For $\mathcal{P} = (d_1 + d_2)/2$ and $\mathcal{P} = \sqrt{d_1 \cdot d_2}$, the results from the extended LIA are not the same as the actual α -cuts because now the boundaries of the actual α -cuts are not parallel to the boundaries of the sub-rectangle. But the the results from the extended LIA approximate the actual α -cuts with good accuracy.

For the problem with multiple performance variables, the same dividing method is applied to the modified $\vec{f}^*(d_1, d_2)$ as the one used in the beginning of this section to demonstrate the extension of the LIA with linear approximations and optimizations. In short, the result of the new extension of the LIA with \mathcal{S} design variable subregions and \mathcal{U} performance variable subregions is denoted as $P_{\alpha_k}^{d\Box}(\mathcal{S}, \mathcal{U})$. Also in order to show the effects of the changes in \mathcal{S} and \mathcal{U} , results from different values of \mathcal{S} and \mathcal{U} are shown in Figure 4.9. Because of the anticipated complexity of the $\vec{f}^*(d_1, d_2)$, \mathcal{U} is chosen as $2\mathcal{S}$. The mark “ \times ” in each figure is $\vec{p}_\times = \vec{f}^*(-0.88, 0.48)$ and it can be seen $\vec{p}_\times \notin P_{\alpha_k}^{d\Box}$.

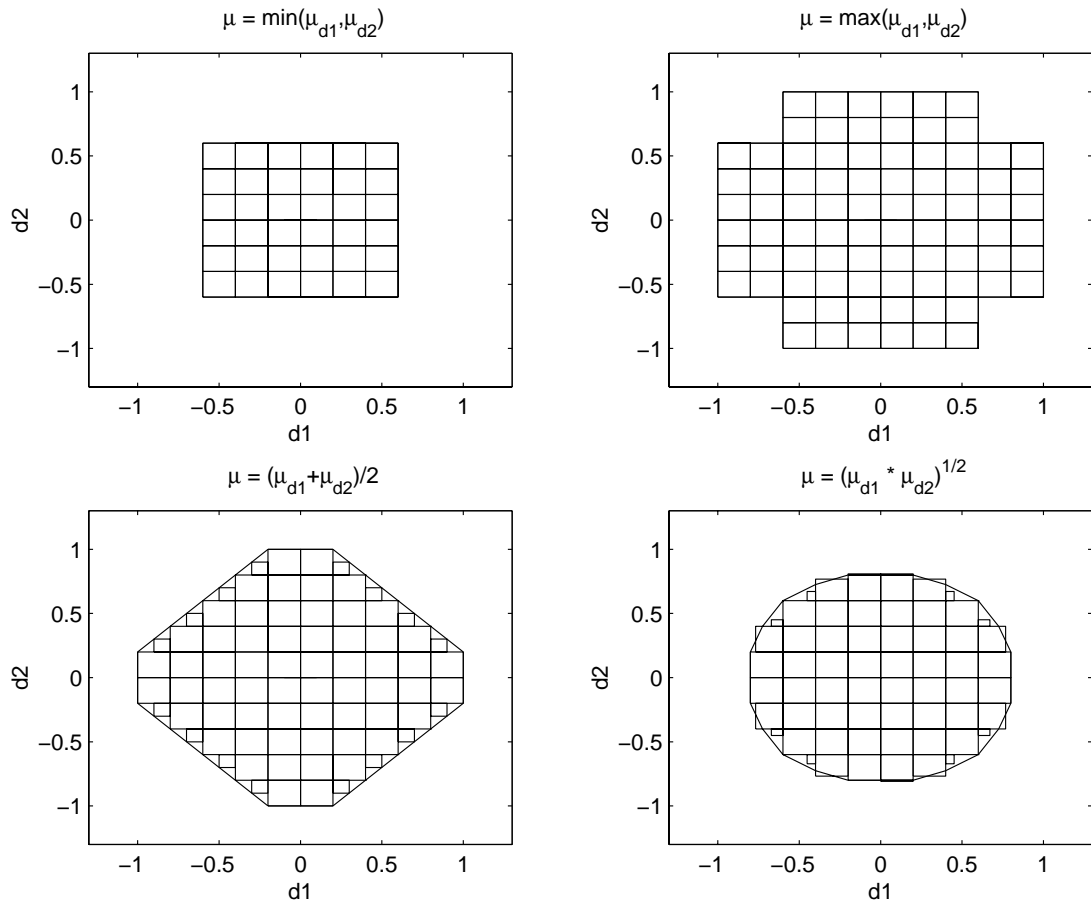


Figure 4.8: $D_{0.5}^d(10)$'s by different aggregation functions.

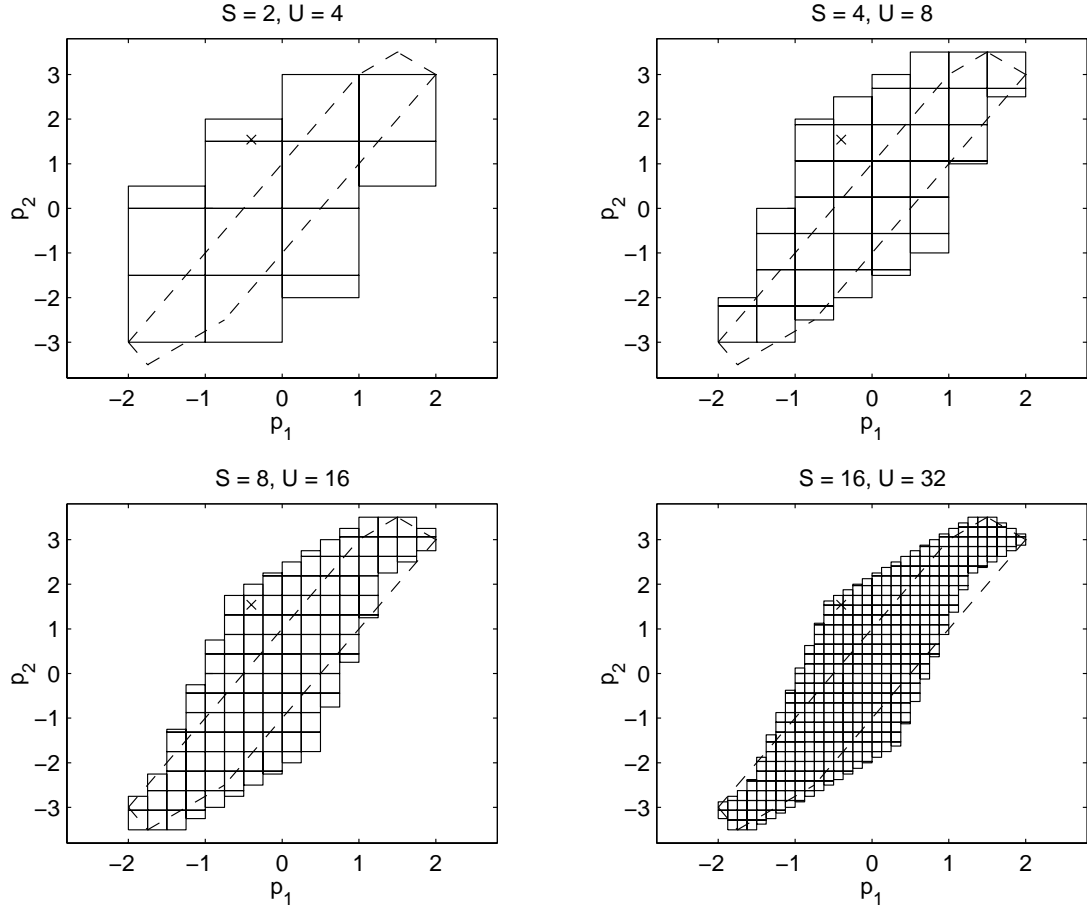


Figure 4.9: $P_{\epsilon}^{d\Box}(S, U)$ with different values of T and U .

but $\vec{p}_{\times} \in P_{\alpha_k}^{d\Box}(S, 2S)$, $S = 2, 4, 8, 16$. If $P_{\alpha_k}^{d\Box}(16, 32)$ is considered as the actual α -cut, it can be seen that even $P_{\alpha_k}^{d\Box}(2, 4)$ is much better than $P_{\alpha_k}^{d\sharp}$, and $P_{\alpha_k}^{d\Box}(4, 8)$ already has acceptable accuracy, and with almost no big difference between $P_{\alpha_k}^{d\Box}(8, 16)$ and $P_{\alpha_k}^{d\Box}(16, 32)$.

$P_{\alpha_k}^{d\Box}(S, U)$ is better than $P_{\alpha_k}^{d\sharp}$ and $P_{\alpha_k}^{d\sharp}$ with respect to flexibility of aggregation function and dimensionality of the PVS. The anomaly that $P_{\alpha_k}^{d\Box} \subset P_{\alpha_l}^{d\Box}$ where $\alpha_k < \alpha_l$ may still be triggered by the nonmonotonicity of $\vec{f}(\vec{d})$ although the chances can be reduced by dividing the DVS and the PVS. To solve this problem, the following operation for multi- α -cuts with k increasing from 1 to $M - 1$ can be added:

$$P_{\alpha_{k+1}}^{d\Box} = P_{\alpha_{k+1}}^{d\Box} \cup P_{\alpha_k}^{d\Box}, \quad \forall 1 \leq k \leq M - 1 \quad (4.16)$$

where $\alpha_k < \alpha_l, \quad \forall 1 \leq k < l \leq M$

One thing that should be noticed is that this new extension of the LIA can be used to induce any preference function from the DVS to the PVS. The new extension of the LIA improves the accuracy at the expense of the increase in the computational cost. If the metamodel of the mapping function is used, the increased computational cost is reasonable as demonstrated in Section 5.3.1.

4.4 The Computation of the Overall Preference

With the extension of the LIA in Section 4.3, now preferences can be aggregated in the DVS and accurately mapped to any dimensional PVS with any aggregation function. The next step in the method of imprecision is to compute the α -cuts of the overall preferences in the DVS and the PVS, $D_{\alpha_k}^o$ and $P_{\alpha_k}^o$.

Otto, Lewis, and Antonsson [39] proved that the maximum overall preference in the DVS is the same as that in the PVS, $\max \mu_o(\vec{d}) = \max \mu_o(\vec{p}) = \mu_o^*$, and $\vec{f}(D_{\mu_o^*}^o) = P_{\mu_o^*}^o$, if \vec{f} and $\mu_d(\vec{d})$ satisfy some continuity conditions. Based on these results, Law *et al.* developed a two-step method to compute $D_{\mu_o^*}^o$ and $P_{\mu_o^*}^o$, and demonstrated this method on the design of a turbofan engine, where the $p = f(\vec{d})$ is the design cost which is generated by the Engine Development Cost Estimator provided by the General Electric Aircraft Engines [27, 28].

- The first step of this method is called the “Forward Calculation,” which is:

- compute $\mu_d(\vec{p})$ from $\mu_d(\vec{d})$ by LIA
- compute $\mu_o(\vec{p}) = \mathcal{P}(\mu_d(\vec{d}), \mu_p(\vec{d}))$
- find $\mu_o^* = \max \mu_o(\vec{p})$ and $P_{\mu_o^*}^o$

- The second step is called “Backward Path,” which is

- find $D_{\mu_o^*}^o$ from $P_{\mu_o^*}^o$ with the help of \vec{f}

The forward calculation is covered by previous sections and chapters. This section focuses on the backward path. In general, \vec{f}^{-1} is needed to map $P_{\mu_o^*}^o$ onto the DVS in order to get $D_{\mu_o^*}^o$, but the problem is that usually \vec{f} is not invertible. For the turbofan engine design problem, there are three special conditions which are helpful to avoid the difficulties caused by \vec{f}^{-1} . First, all eight design variables are discrete. So the DVS is a finite set of design alternatives, then $D_{\mu_o^*}^o$ can be found by brute-force search. Second, $\mathcal{P}(\mu_d, \mu_p) = \min(\mu_d, \mu_p)$, which means that $\mu_d(\vec{d}) \geq \mu_o(\vec{d})$, $\forall \vec{d} \in DVS$.

Then $\forall \alpha_k \leq \mu_o^*$, $D_{\alpha_k}^d \supseteq D_{\mu_o^*}^o$, because $\mu_d(\vec{d}^*) \geq \mu_o^* \geq \alpha_k, \forall \vec{d}^* \in D_{\mu_o^*}^o$. Because the larger the α_k , the smaller the size of $D_{\alpha_k}^d$, the search space of $D_{\mu_o^*}^o$ is now restricted to $D_{\alpha_k^*}^d$ where α_k^* is the largest α_k among all $\alpha_k \leq \mu_o^*$.

Although the backward path succeeded for the engine design problem, there are some difficulties for a general problem without the three special conditions. First, if there is compensation in the aggregation function of μ_d and μ_p , the search space of $D_{\mu_o^*}^o$ is the whole DVS. Second, the brute-force search is not feasible if the DVS is not a finite set of design alternatives. When there is compensation in the aggregation function of μ_d and μ_p , the α -cut of μ_o may not be limited to the α -cut of μ_d . There are two ways to compute the α -cut of μ_o from the α -cut of μ_d : either by intersection of $P_{\alpha_k}^d$ and $P_{\alpha_k}^p$ in which way some region of $P_{\alpha_k}^o$ is lost, or reconstruct $\mu_d(\vec{p})$ from $P_{\alpha_k}^d$'s which is not trivial. Finally, if $\vec{f}(\vec{d})$ is nonmonotonic, it is possible that $\vec{f}^{-1}(P_{\mu_o^*}^o) \supset D_{\mu_o^*}^o$, where \vec{f}^{-1} is not a single-valued function and defined as

$$\vec{f}^{-1}(P) = \{\vec{d} \in \mathcal{X} \mid \vec{f}(\vec{d}) = \vec{p}, \vec{p} \in P\} \quad \forall P \subseteq \mathcal{Y} \quad (4.17)$$

where \mathcal{Y} is the set of all vectors of performance variables with valid values.

Because $\vec{f}(\vec{d})$ is nonmonotonic, $\exists d_1, d_2 \in DVS$ such that $\vec{f}(d_1) = \vec{f}(d_2) = \vec{p}_*$. Without loss of generality, assume that $\mu_d(d_2) > \alpha_2 > \mu_d(d_1) > \alpha_1 > 0$. According to the extension principle, $\mu_d(\vec{p}_*) \leq \max(\mu_d(\vec{d}_1), \mu_d(\vec{d}_2)) = \mu_d(\vec{d}_2) > \alpha_2$. Assume $\alpha_2 > \mu_p(\vec{p}_*) > \alpha_1$, then there exists $\mathcal{P}_* \neq \max$ such that $\mu_o(\vec{p}_*) = \mathcal{P}_*(\mu_d(\vec{p}_*), \mu_p(\vec{p}_*)) > \alpha_2$, i.e.:

$$\vec{p}_* \in P_{\alpha_2}^o \quad \text{and} \quad \vec{d}_1, \vec{d}_2 \in \vec{f}^{-1}(P_{\alpha_2}^o) \quad (4.18)$$

But because $\mu_d(\vec{d}_1) < \alpha_2$ and $\mu_p(\vec{p}_*) < \alpha_2$, from the idempotency and monotonicity of \mathcal{P} , $\mu_o(\vec{d}_1) = \mathcal{P}(\mu_d(\vec{d}_1), \mu_p(\vec{f}(\vec{d}_1))) = \mathcal{P}(\mu_d(\vec{d}_1), \mu_p(\vec{p}_*)) < \mathcal{P}(\alpha_2, \alpha_2) = \alpha_2$, which means that

$$\vec{d}_1 \notin D_{\alpha_2}^o \quad (4.19)$$

From Equation 4.18 and Equation 4.19, it can be concluded that $\vec{f}^{-1}(P_{\mu_o^*}^o) \supset D_{\mu_o^*}^o$ for aggregation function \mathcal{P}_* and $\mu_o^* > \alpha_2$.

All above difficulties and errors stem from the usage of \vec{f}^{-1} . They can be avoided if \vec{f}^{-1} is no longer needed. The following method can compute $D_{\alpha_k}^o$ and $P_{\alpha_k}^o$ without \vec{f}^{-1} , and so it will produce more accurate results with less computational cost.

- The new method to compute $D_{\alpha_k}^o$ and $P_{\alpha_k}^o$ without \vec{f}^{-1} :
 1. Specify \vec{d} , \vec{p} , \mathcal{X} , \mathcal{Y} , $\mu_{d_i}(d_i)$'s, and $\mu_{p_i}(p_i)$'s. Decide the trade-off strategy, *i.e.*, the \mathcal{P} used to compute the overall preference $\mu_o = \mathcal{P}(\mu_d, \mu_p)$.
 2. Compute $\mu_o(\vec{d})$ the overall preference in DVS, and use the method in Section 4.3 to generate $D_{\alpha_k}^{o\Box}$.
 3. Compute $P_{\alpha_k}^{o\Box}$ with $D_{\alpha_k}^{o\Box}$ by the extension of the LIA in Section 4.3.

The biggest difference between the new method and the old method is that the new method does not need \vec{f}^{-1} . In the old method computations are carried out in both the DVS and the PVS, and finally $P_{\alpha_k}^o$ is generated before $D_{\alpha_k}^o$, which is why it needs \vec{f}^{-1} . But in the new method, although $\mu_p(\vec{f}(\vec{d}))$ can be considered computations in the PVS, the final aggregation for the overall preference is in DVS. Computing $\mu_p(\vec{f}(\vec{d}))$ for $\vec{d} \in \mathcal{X}$ is the basis of the new method. The LIA is a discrete approximate implementation of the extension method. When apply LIA to $D_{\alpha_k}^{o\Box}$, $\mu_o(\vec{d})$ is needed to be computed only at a finite number of points in DVS. Even if \vec{f} is not invertible over the PVS but it can be considered invertible at any single point in the PVS because the \vec{f}^{-1} in Equation 4.17 is single-valued for $P = \{\vec{f}(\vec{d})\} \subseteq \mathcal{Y}$. The new method can be understood as applying the two-step old method at a finite number of design points where \vec{f}^{-1} is well defined. The usage of \vec{f}^{-1} is avoided by transferring the information of the functional requirements in the the PVS to the DVS by using the equivalent inverse of \vec{f} at individual points in the PVS.

To demonstrate the new method, it will be applied to the example in Section 4.3 with a two-dimensional DVS and a two-dimensional PVS. The design preferences for both design variables are defined in Figure 4.1. The performance function is shown in Equation 4.9. The design preferences and functional requirements are shown in Figure 4.10. The aggregation functions for the combined design preference, combined functional requirement, and the overall preference in the DVS are

$$\begin{aligned}
 \mu_d(\vec{d}) &= \min(\mu_{d_1}(d_1), \mu_{d_2}(d_2)) \\
 \mu_p(\vec{d}) &= [\mu_{p_1}(f_1(\vec{d})) + \mu_{p_2}(f'_2(\vec{d}))] / 2 \\
 \mu_o(\vec{d}) &= \sqrt{\mu_d(\vec{d}) \cdot \mu_p(\vec{d})}
 \end{aligned} \tag{4.20}$$

The overall preference in DVS is shown in Figure 4.11. Here $\alpha_k = 0.5$ is chosen, because the α -cut will be more complicated than those for $\alpha_k = \varepsilon$ or 1.0.

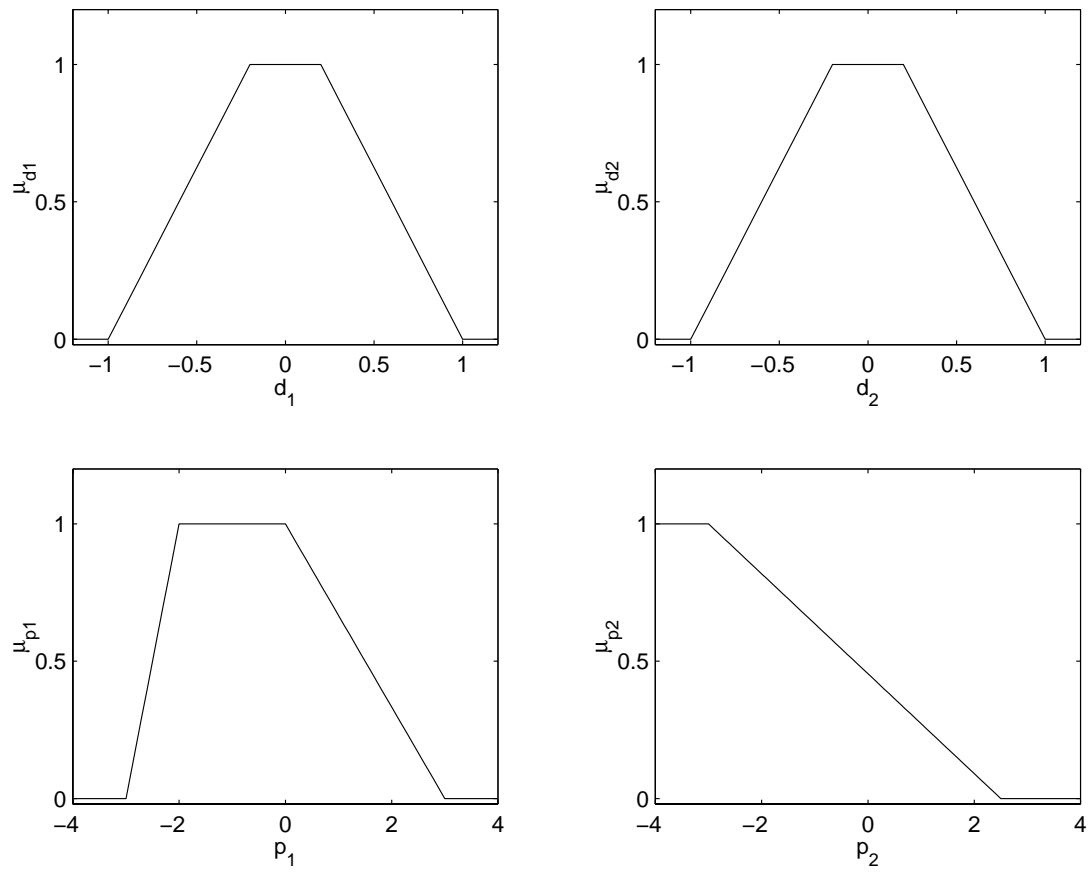


Figure 4.10: $\mu_{d_1}(d_1)$, $\mu_{d_2}(d_2)$, $\mu_{p_1}(p_1)$ and $\mu_{p_2}(p_2)$.

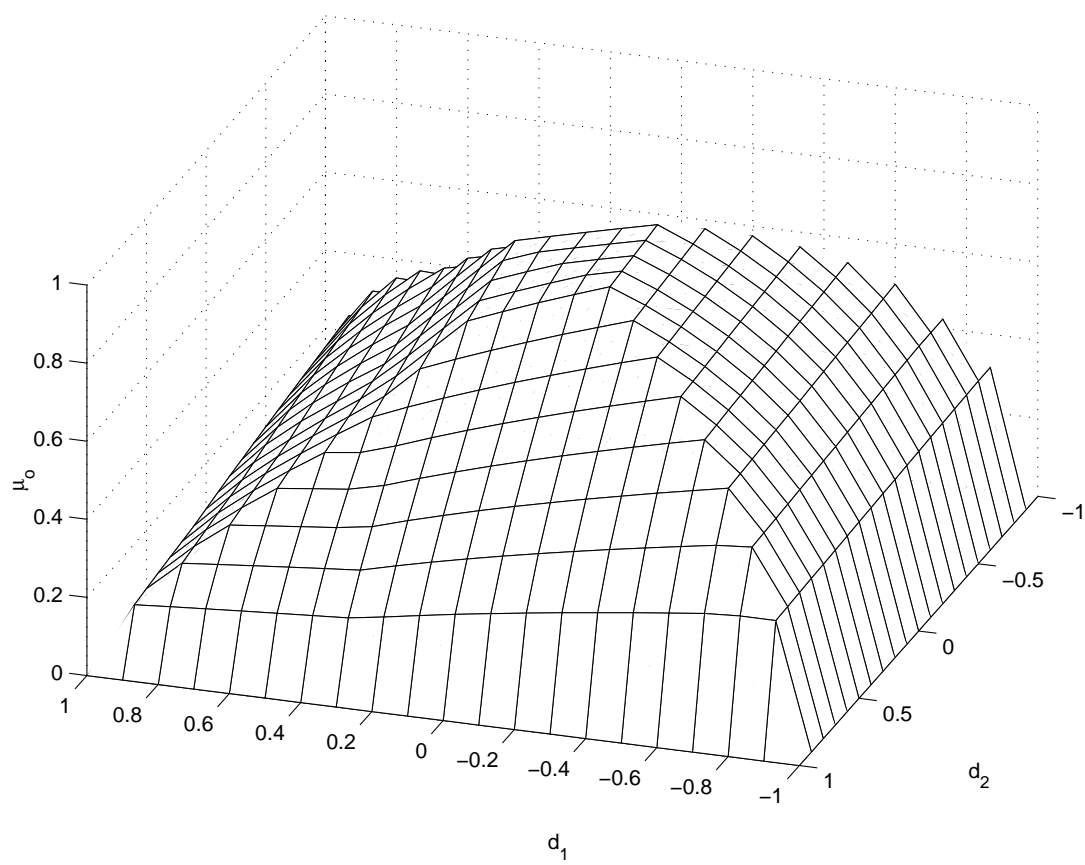


Figure 4.11: The shape of $\mu_o(\vec{d})$.

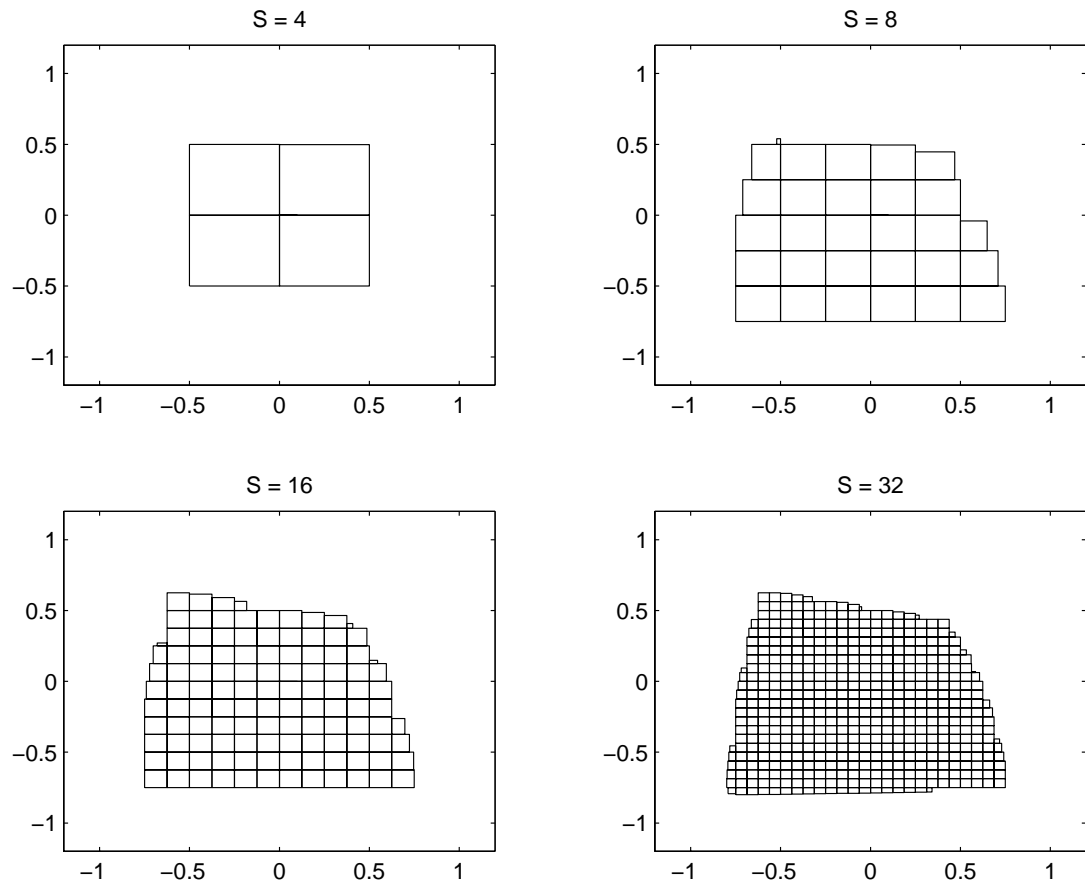


Figure 4.12: $D_{0.5}^{\circ\Box}(\mathcal{S})$ for different values of \mathcal{S} .

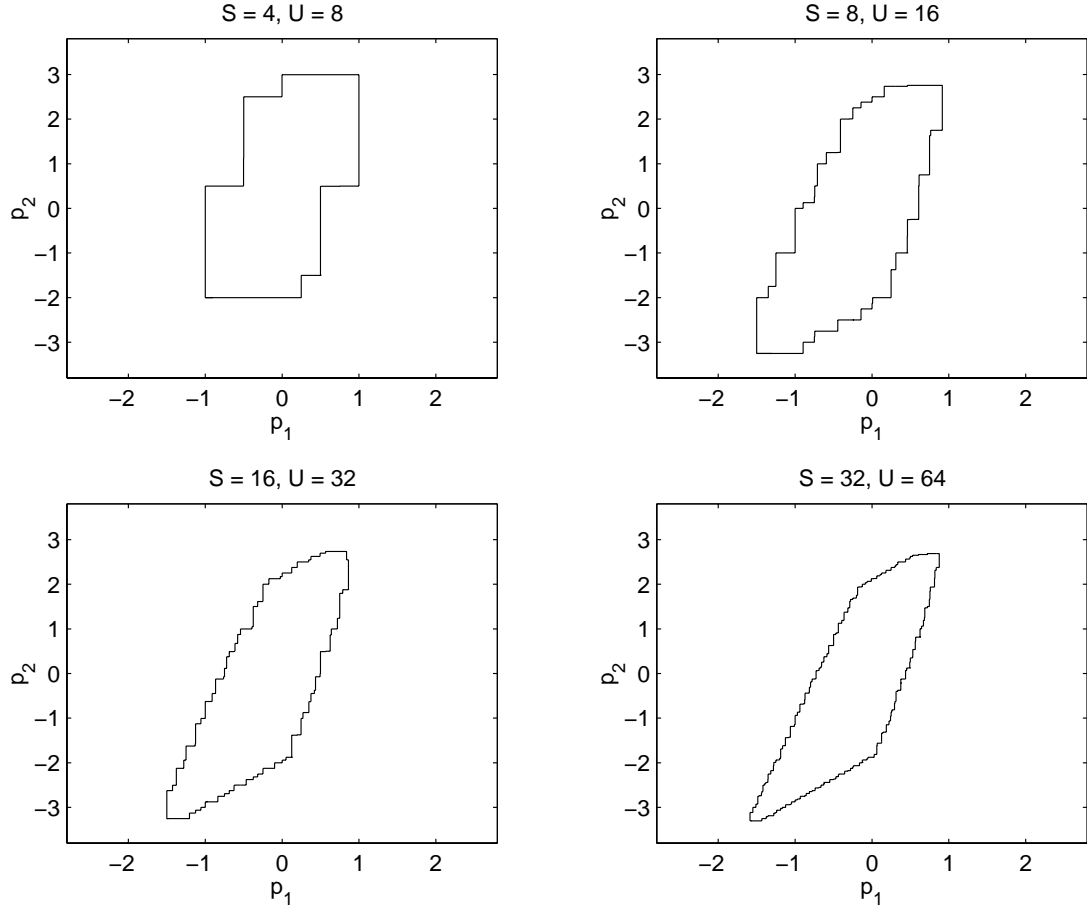


Figure 4.13: $P_{0.5}^{o\Box}(S, U)$ for different values of S and U .

Now the new method is used to compute $D_{0.5}^{o\Box}$ and $P_{0.5}^{o\Box}$. Following procedure is listed in Section 4.4, $D_{0.5}^{o\Box}$ should be computed first. Here different T 's are chosen as 4, 8, 16 and 32 to compare the effect of S . These $D_{0.5}^{o\Box}$'s are shown in Figure 4.12. Among the four $D_{0.5}^{o\Box}$'s, $D_{0.5}^{o\Box}(32)$ is of course the most accurate, and $D_{0.5}^{o\Box}(16)$ also catches many details in $D_{0.5}^{o\Box}(32)$, even $D_{0.5}^{o\Box}(8)$ has enough details for the use in preliminary stages. The $P_{0.5}^{o\Box}$'s are also induced with $U = 2S$ as shown in Figure 4.13. Not surprisingly, $P_{0.5}^{o\Box}(32, 64)$ is the best one. And the $P_{0.5}^{o\Box}(16, 32)$ is almost the same as $P_{0.5}^{o\Box}(32, 64)$. $P_{0.5}^{o\Box}(8, 16)$ is the most cost-effective result among these four $P_{0.5}^{o\Box}$'s. Although $D_{0.5}^{o\Box}(4)$ is rough, $P_{0.5}^{o\Box}(4, 8)$ is still on the right track if compared with others.

Now the DVS is continuous, and \vec{f} is not invertible over the whole DVS. The backward path can not be used to compute $D_{0.5}^o$. So the result of the new method and the result of the forward calculation can be compared. $(S, U) = (8, 16)$ is chosen because it is the cost-effective setting. $\tilde{P}_{0.5}^{d\Box}(8, 16)$ and $\tilde{P}_{0.5}^{o\Box}(8, 16)$ are computed by the forward calculation and are shown in Figure 4.14.

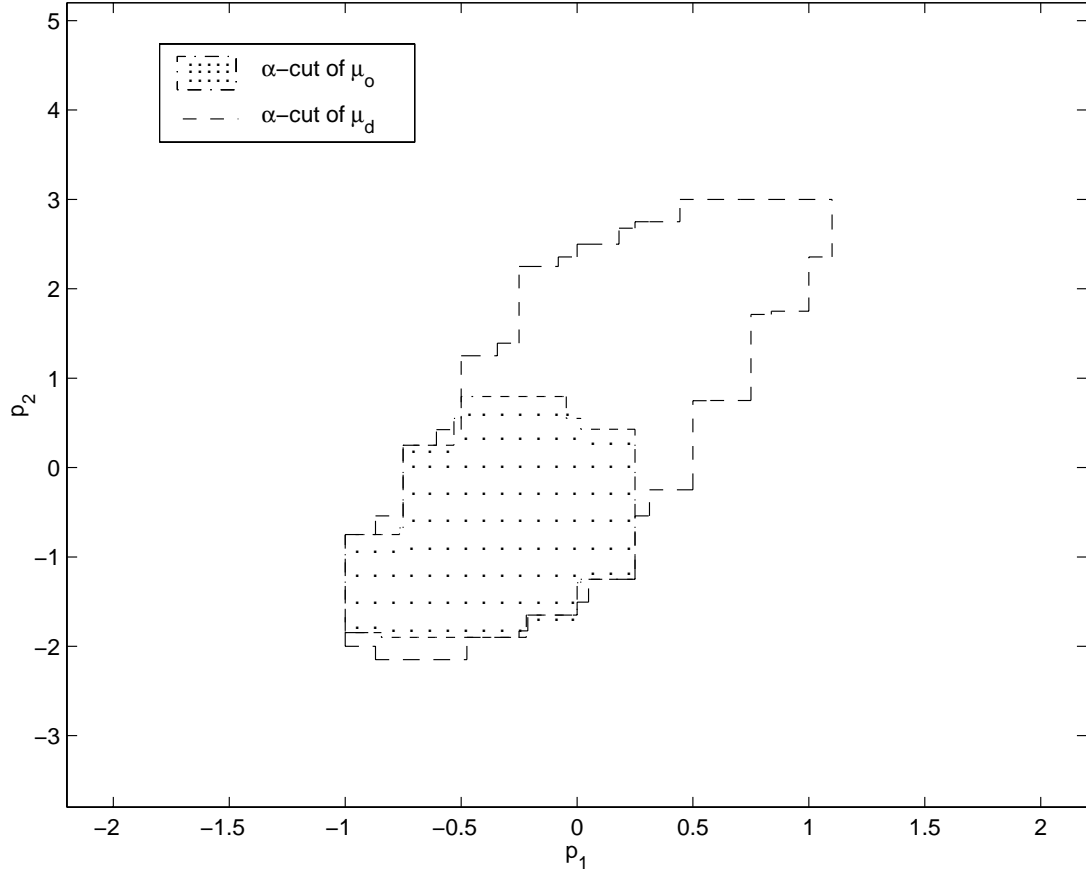


Figure 4.14: $P_{0.5}^{d\Box}(8, 16)$ and $P_{0.5}^{o\Box}(8, 16)$ by the forward calculation.

In order to simplify the computation, only the subregions within $\tilde{P}_{0.5}^{d\Box}(8, 16)$ are used to compute $\tilde{P}_{0.5}^{o\Box}(8, 16)$, although there is compensation in $\mu_o = \sqrt{\mu_d \cdot \mu_p}$.

Figure 4.15 shows the $P_{0.5}^{o\Box}(8, 16)$ and the $\tilde{P}_{0.5}^{o\Box}(8, 16)$, along with the α -cut generated by the forward calculation with the original LIA, which can be considered as $\tilde{P}_{0.5}^{o\Box}(1, 1)$. Several test points are also drawn in Figure 4.15, where $\vec{p}_\times = \vec{f}'(\vec{d}_\times) = \vec{f}'(-0.2, 0.2)$, $\vec{p}_* = \vec{f}'(\vec{d}_*) = \vec{f}'(0.6, 0.4)$, and $\vec{p}_+ = \vec{f}'(\vec{d}_+) = \vec{f}'(0.48, 0.36)$. $\mu_o(\vec{p}_\times) \geq \mu_o(\vec{d}_\times) = 0.79201 > \alpha = 0.5$ indicates the error of $\tilde{P}_{0.5}^{o\Box}(8, 16)$ in the region around \vec{p}_\times which may be introduced by not including points outside $\tilde{P}_{0.5}^{d\Box}(8, 16)$. $\mu_o(\vec{d}_*) = 0.40825 < \alpha = 0.5$ indicates possible error of $\tilde{P}_{0.5}^{o\Box}(1, 1)$ in the region around \vec{p}_* . And $\mu_o(\vec{d}_+) = 0.40825 < \alpha = 0.5$ indicates possible error of $P_{0.5}^{o\Box}(8, 16)$ at its upper-right corner around \vec{p}_+ .

Among these three α -cuts of the overall preference at α -level 0.5, the result of the forward calculation with the original LIA, $\tilde{P}_{0.5}^{o\Box}(1, 1)$, is the least accurate one. If the LIA was replaced by the extension with 8 subregions in each DV and 16 subregions in each PV, the α -cut was refined.

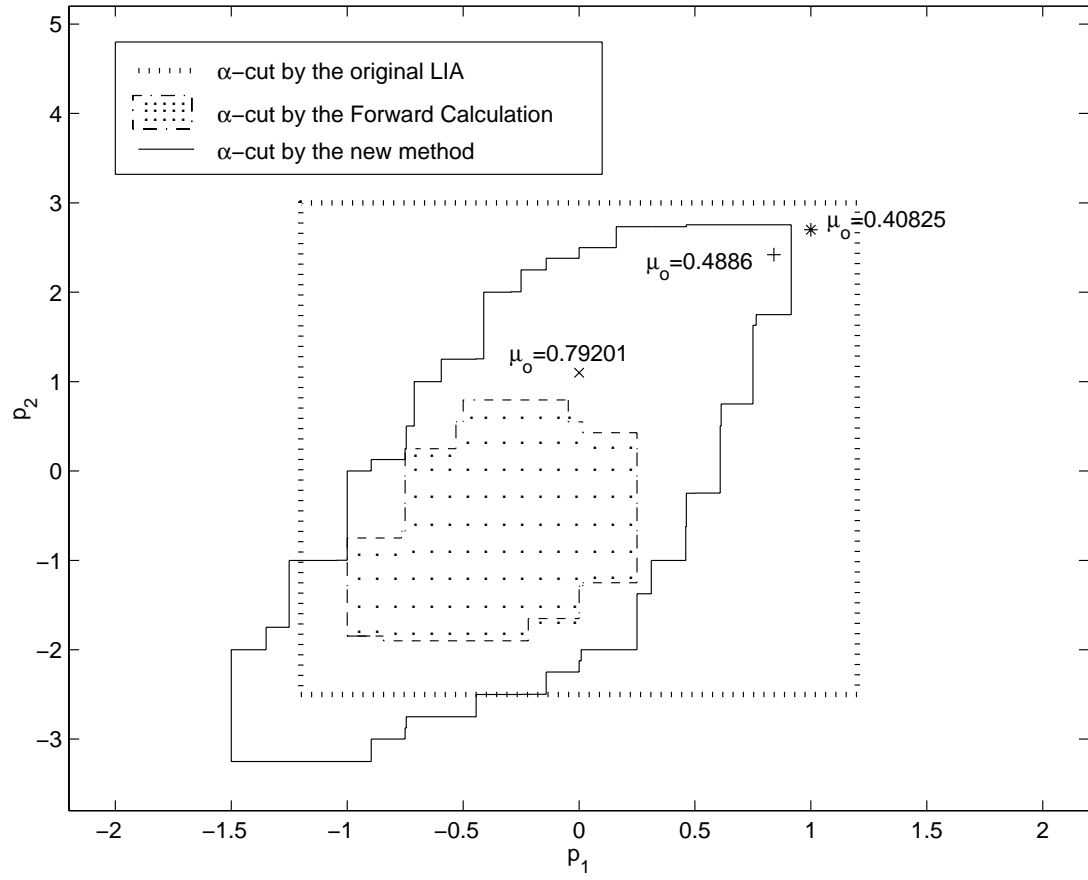


Figure 4.15: $P_{0.5}^{o\Box}(8, 16)$ by the new method and $P_{0.5}^{o\Box}(1, 1)$ and $P_{0.5}^{o\Box}(8, 16)$ by the forward calculation.

But because $\mu_o(\vec{p})$ was not calculated for $\vec{p} \notin \tilde{P}_{0.5}^{d\Box}(1, 1)$, $\tilde{P}_{0.5}^{o\Box}(8, 16)$ will be smaller than the actual $P_{0.5}^o$. For the resulting α -cut by the new method $P_{0.5}^{o\Box}(8, 16)$, there are errors around its boundary because $\mathcal{S} = 8$ and $\mathcal{U} = 16$. Increasing \mathcal{S} and \mathcal{U} can increase the accuracy of $P_{0.5}^{o\Box}$.

4.5 Summary

If the new extension of the LIA is added to the new method to compute the α -cuts in the DVS and the PVS, it becomes a new method which can compute the α -cuts of the overall preference with much higher accuracy. The procedure of this implementation in detail is

1. Specify \vec{d} , \vec{p} , \mathcal{X} , \mathcal{Y} , $\mu_{d_i}(d_i)$'s, and $\mu_{p_i}(p_i)$'s. Decide the trade-off strategy, *i.e.*, the \mathcal{P} used to compute the overall preference $\mu_o = \mathcal{P}(\mu_d, \mu_p)$.
2. Divide the relevant range of each d_i into subregions \mathcal{X}_{i,r_i} , which form $\mathcal{X}_{\vec{r}}^*$ the sub-hypercubes in the DVS.
3. Construct $\mu_{o,\vec{r}}(d_i)$, the local overall preference in each sub-hypercube.
4. Find the $D_{\vec{r},\alpha_k}^o$, the α -cut of the local overall preference in each sub-hypercube.
5. The union of the $D_{\vec{r},\alpha_k}^o$'s in all the sub-hypercubes in DVS is $D_{\alpha_k}^{o\Box}$, the whole α -cut of the global overall preference in DVS.
6. Divide the relevant range of each p_j into subregions \mathcal{Y}_{j,t_j} , which form $\mathcal{Y}_{\vec{t}}^*$, the sub-hypercubes in the PVS.
7. For each $\mathcal{Y}_{\vec{t}}^*$, find the interval of each $D_{\vec{r},\alpha_k}^o$ on p_j by the original LIA, whose union is the interval of the global overall preference on p_j in $\mathcal{Y}_{\vec{t}}^*$.
8. $P_{\vec{t},\alpha_k}^o$, the α -cut of the global overall preference μ_o in each sub-hypercube, is the Cartesian product of the intervals of μ_o on p_j in $\mathcal{Y}_{\vec{t}}^*$.
9. The union of the $P_{\vec{t},\alpha_k}^o$'s in all the sub-hypercubes in the PVS is $P_{\alpha_k}^{o\Box}$, the whole α -cut of the global overall preference in PVS.

The above method can be used to compute the α -cut of the overall preference in the DVS and the PVS for any α level between ε and 1.0. If there are several α levels in ascending order, $\{\alpha_1, \dots, \alpha_M\}$, the operations in Equation 4.17 should be added after the step 4 in the above list.

The increase of the computational cost is reasonable if the metamodel of the mapping function is used, which will be demonstrated in Section 5.3.1.

Dividing the DVS into sub-hypercubes and constructing local overall preference function reduces the error caused by the aggregation functions. If there are local extrema of \vec{f} , the anomaly will only affect the sub-hypercubes containing the local extrema. Dividing the PVS into sub-hypercubes can relax the requirement about the independence of f_j 's. Now it only requires that the f_j 's are independent within each sub-hypercube. Even if the independence assumption is violated, the error is also reduced and limited within the mapping of that sub-hypercube. The existence of \vec{f}^{-1} is no longer necessary because of the change of the computation order of $D_{\alpha_k}^{o\Box}$ and $P_{\alpha_k}^{o\Box}$.

Chapter 5

Implementation of the M_OI and Example

The basic definitions, the construction of the metamodel of the mapping function, the revised extension of the LIA, and a new method to compute the overall preference are discussed in previous chapters. This chapter will introduce the implementation of the M_OI which combines all techniques mentioned above. A new measure will be proposed to test the improvement of metamodels. This implementation will be demonstrated on the design of the structure of a passenger vehicle.

5.1 Implementation of the M_OI

Chapter 4 discussed how to improve the accuracy of the LIA and how to compute the overall preference more accurately without consideration for the computation cost of $\vec{f}(\vec{d})$. A less expensive metamodel of $\vec{f}(\vec{d})$ should be used when $\vec{f}(\vec{d})$ is prohibitively expensive to compute. The model structure of the metamodel for $f_j(\vec{d})$, and the sampling criteria for multistage metamodels, are discussed in Chapter 3. The empirical root-mean-square error, the *ERMSE*, was used to evaluate the performance of the metamodels. In the implementation of the M_OI , a new single measure for the metamodel performance instead of the *ERMSE* is preferable.

5.1.1 The Difference in the Volumes of α -cuts

With a metamodel \vec{f}'_l of $\vec{f}(\vec{d})$, the results of the M_OI are the α -cuts in the DVS and the PVS, $D_\alpha^{o\Box}$'s and $P_\alpha^{o\Box}$'s. \vec{f}'_l is used to compute $\mu_o(\vec{d})$ and map $D_\alpha^{o\Box}$ onto the PVS. The performance of the metamodel will affect the accuracy of the final results. The *ERMSE* is only a general measure for the performance of an individual interpolation function. In the M_OI , set boundaries (α -cuts) are of more interest than the value of $f'_j(\vec{d})$, so the difference between α -cuts is more meaningful than the

n-tuple of the ERMSE. $P_\alpha^{o\Box}$ is induced from $D_\alpha^{o\Box}$ by the LIA. The LIA will introduce some errors in $P_\alpha^{o\Box}$ because the LIA is an approximate implementation of the extension principle. As discussed above, the accuracy will decrease when some assumptions are violated even if the revised extension of the LIA is used. Hence the difference between $D_\alpha^{o\Box}$'s by different metamodels will be a measure of the metamodel's performance. The procedure to compute this difference is described below.

Consider two metamodels: \vec{f}_l based on the set of design points S_l , and \vec{f}_{l+1} based on the set of design points S_{l+1} . Without loss of generality, it is assumed that $S_l \subset S_{l+1}$. The resulting α -cut with \vec{f}_l is $D_{\alpha_k, l}^{o\Box}$, which consists of the $D_{\vec{r}, \alpha_k, l}^o$'s in all sub-hypercubes. Each $D_{\vec{r}, \alpha_k, l}^o$ is also a n -cube, so it is simple to calculate its volume. First define the volume of the difference between $D_{\vec{r}, \alpha_k, l}^o$ and $D_{\vec{r}, \alpha_k, l+1}^o$:

$$\begin{aligned} \text{Volume}(D_{\vec{r}, \alpha_k, l}^o \cup D_{\vec{r}, \alpha_k, l+1}^o) &= \prod_{i=1}^n \left[\max(d_{i, l+1, \max}^{\alpha_k}, d_{i, l, \max}^{\alpha_k}) - \min(d_{i, l+1, \min}^{\alpha_k}, d_{i, l, \min}^{\alpha_k}) \right] \quad (5.1) \\ \text{Volume}(D_{\vec{r}, \alpha_k, l}^o \cap D_{\vec{r}, \alpha_k, l+1}^o) &= \prod_{i=1}^n \left[\min(d_{i, l+1, \max}^{\alpha_k}, d_{i, l, \max}^{\alpha_k}) - \max(d_{i, l+1, \min}^{\alpha_k}, d_{i, l, \min}^{\alpha_k}) \right] \\ V(D_{\vec{r}, \alpha_k, l+1}^o - D_{\vec{r}, \alpha_k, l}^o) &= \text{Volume}(D_{\vec{r}, \alpha_k, l}^o \cup D_{\vec{r}, \alpha_k, l+1}^o) - \text{Volume}(D_{\vec{r}, \alpha_k, l}^o \cap D_{\vec{r}, \alpha_k, l+1}^o) \end{aligned}$$

The volume of the difference between $D_{\alpha_k, l}^{o\Box}$ and $D_{\alpha_k, l+1}^{o\Box}$ is the sum of the volumes of the difference between each pair of sub- α -cuts:

$$V(D_{\alpha_k, l+1}^{o\Box} - D_{\alpha_k, l}^{o\Box}) = \sum_{\vec{r}} V(D_{\vec{r}, \alpha_k, l+1}^o - D_{\vec{r}, \alpha_k, l}^o) \quad (5.2)$$

Sometimes the ratio of the volume of the difference to the volume of $D_{\alpha_k, l}^{o\Box}$ is preferred to compare the improvements at different α -levels:

$$r_{v_d}(\alpha_k, l, l+1) = \frac{V(D_{\alpha_k, l+1}^{o\Box} - D_{\alpha_k, l}^{o\Box})}{V(D_{\alpha_k, l}^{o\Box})} \quad (5.3)$$

$D_{\alpha_{k1}}^{o\Box} \subseteq D_{\alpha_{k2}}^{o\Box}$, for $\alpha_{k1} > \alpha_{k2}$. If $r_{v_d}(\alpha_{k1}, l, l+1) > r_{v_d}(\alpha_{k2}, l, l+1)$, it can be interpreted that $D_{\alpha_{k1}, l}^{o\Box}$ is more sensitive to the change of \vec{f}_l than $D_{\alpha_{k2}, l}^{o\Box}$. This result suggests that more design points should be sampled in $D_{\alpha_{k1}, l}^{o\Box}$ for the metamodel at the next stage.

5.1.2 Implementation Process

The method discussed in Chapter 4 will only compute one α -cut in the DVS and the PVS. Some modifications are needed if more than one α -level is specified. If \vec{f} is computationally expensive to evaluate, its metamodel \vec{f}' can be used to replace \vec{f} in the method. When multistage metamodels are preferred, the measure r_{vd} can be used to choose the sampling range of design points for the metamodel in the next stage. With all these modifications, the implementation of the M₀I becomes

1. Identify design variables $\{d_1, \dots, d_n\}$ and performance variables $\{p_1, \dots, p_q\}$.
2. Specify design preferences $\{\mu_{d_1}(d_1), \dots, \mu_{d_n}(d_n)\}$ and functional requirements $\{\mu_{p_1}(p_1), \dots, \mu_{p_q}(p_q)\}$.
3. Decide the aggregation hierarchy for the overall preference and parameters of each aggregation function.
4. Specify interesting α -levels $\{\alpha_1, \dots, \alpha_M\}$ where $\alpha_k < \alpha_{k+1}, 1 \leq k \leq M - 1$.
5. Let $l = 1$. Decide the design points S_1 for the first metamodel \vec{f}'_1 by experimental design.
6. Repeat until satisfactory accuracy is achieved:
 - (a) Build the l^{th} metamodel \vec{f}'_l of \vec{f} with design points S_l .
 - (b) Use \vec{f}'_l to compute $D_{\alpha_k}^{o\Box}$ for $1 \leq k \leq M$.
 - (c) If $l \geq 2$, then compute $r_{vd}(\alpha_k, l - 1, l)$ for $1 \leq k \leq M$, and find $D_{\alpha_s}^{o\Box}$ which is most sensitive to the change of metamodels. Otherwise let $s = 1$.
 - (d) Sample a set of points, dS_{l+1} , for \vec{f}'_{l+1} within the bounding hypercube of $D_{\alpha_s}^{o\Box}$, but outside the bounding hypercube of $D_{\alpha_{s+1}}^{o\Box}$ if $s < M$, then let $S_{k+1} = S_k \cup dS_{l+1}$.
 - (e) $l = l + 1$.
7. Compute $P_{\alpha_k}^{o\Box}$ for $1 \leq k \leq M$ with the last metamodel, \vec{f}'_{l-1} .

The whole procedure will be demonstrated in the following sections.

5.2 Problem Description

The problem used in the example is the preliminary vehicle structural design of a 1980 VW Rabbit, which was an application of the M₀I demonstrated to Volkswagen Wolfsburg in the summer of



Figure 5.1: Testing setup of body-in-white.

1997 [51]. In general, the vehicle structural design is to optimize the *body-in-white* of a vehicle, which is the portion of the body of the vehicle that carries the loads. The design engineer needs to refine the design in order to meet some quantified engineering targets such as stiffness or weight. Other *soft* specifications such as style or manufacturability, which are not easy to quantify, are also be taken into consideration.

The bending stiffness and the torsional stiffness were measured for the body-in-white. The 1980 VW Rabbit has a bending stiffness of approximately 2500 N/mm and a torsional stiffness of approximately 4900 N-m/degree. A solid model and a finite element model were created from the geometry data measured from the body-in-white.

5.2.1 Design Variables and Performance Variables

The problem here is to improve the overall performance 10%. There are five design variables which will represent the manufacturability and style:

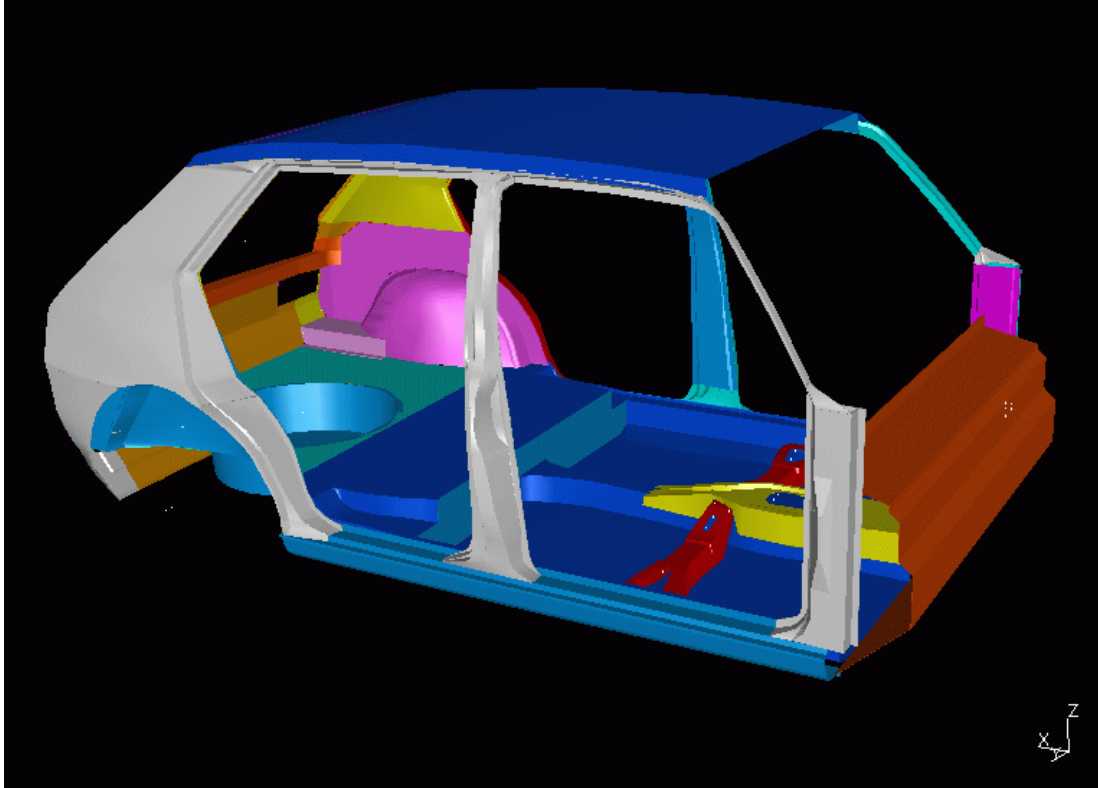


Figure 5.2: Geometric model of body-in-white in SDRC I-DEAS.

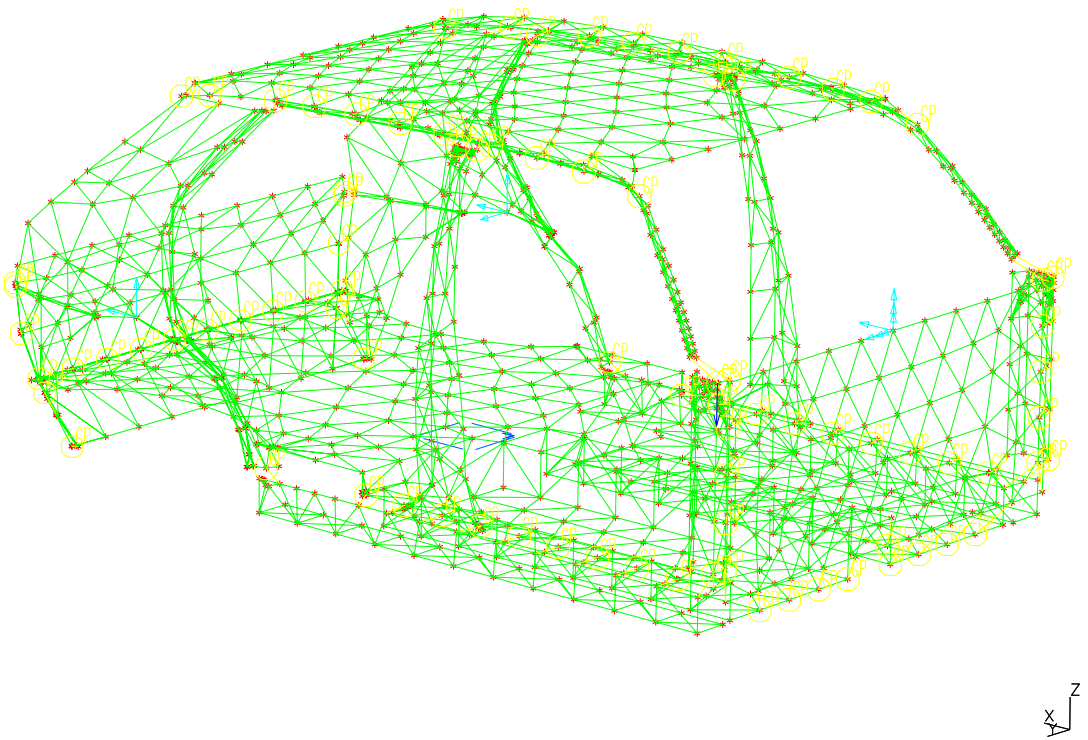


Figure 5.3: Finite element model of body-in-white.

d_1 = A Pillar Thickness [mm]

d_2 = B Pillar Thickness [mm]

d_3 = Floor Rail Thickness [mm]

d_4 = Floor Thickness [mm]

d_5 = B Pillar Location [mm]

and three performance variables:

p_1 = Bending Stiffness [N/mm]

p_2 = Torsional Stiffness [N-m/deg]

p_3 = Weight [kg]

5.2.2 Design Preferences and Functional Requirements

The design preferences obtained from engineers and stylists [51] are shown in Figure 5.4. Because the sheet steel used to build A-pillar is only available in certain thickness, $\mu_{d_1}(d_1)$ is obtained only at certain d_1 values. Thinner sheet steel is easier to form, so $\mu_{d_1}(d_1)$ represents the manufacturability. The design preference for the B-pillar thickness is continuous because of the simplifications in the finite element model. A thicker B-pillar requires more reinforcing features, so a thinner and simpler B-pillar has higher preference. The preference of the floor thickness reflects the availability of materials with such thickness. A thicker floor pan is easier to attach and more durable, so it has higher preference. The preference of the B-pillar location is for the vehicle's style. It is specified in α -cuts at three α -levels, $\{\varepsilon, 0.5, 1.0\}$. The continuous version of design preferences is built by connecting discrete points with lines for μ_{d_1} , μ_{d_3} , μ_{d_4} and μ_{d_5} .

The functional requirements are gathered from customers or managers by asking what is the extreme value of a performance variable while the performance is acceptable, $\mu_p \geq \varepsilon$, or the performance is ideal, $\mu_p = 1.0$. Then these pairs of points are connected by straight lines. So they are in simpler form as piecewise linear functions [51]. All three functional requirements are shown in Figure 5.5.

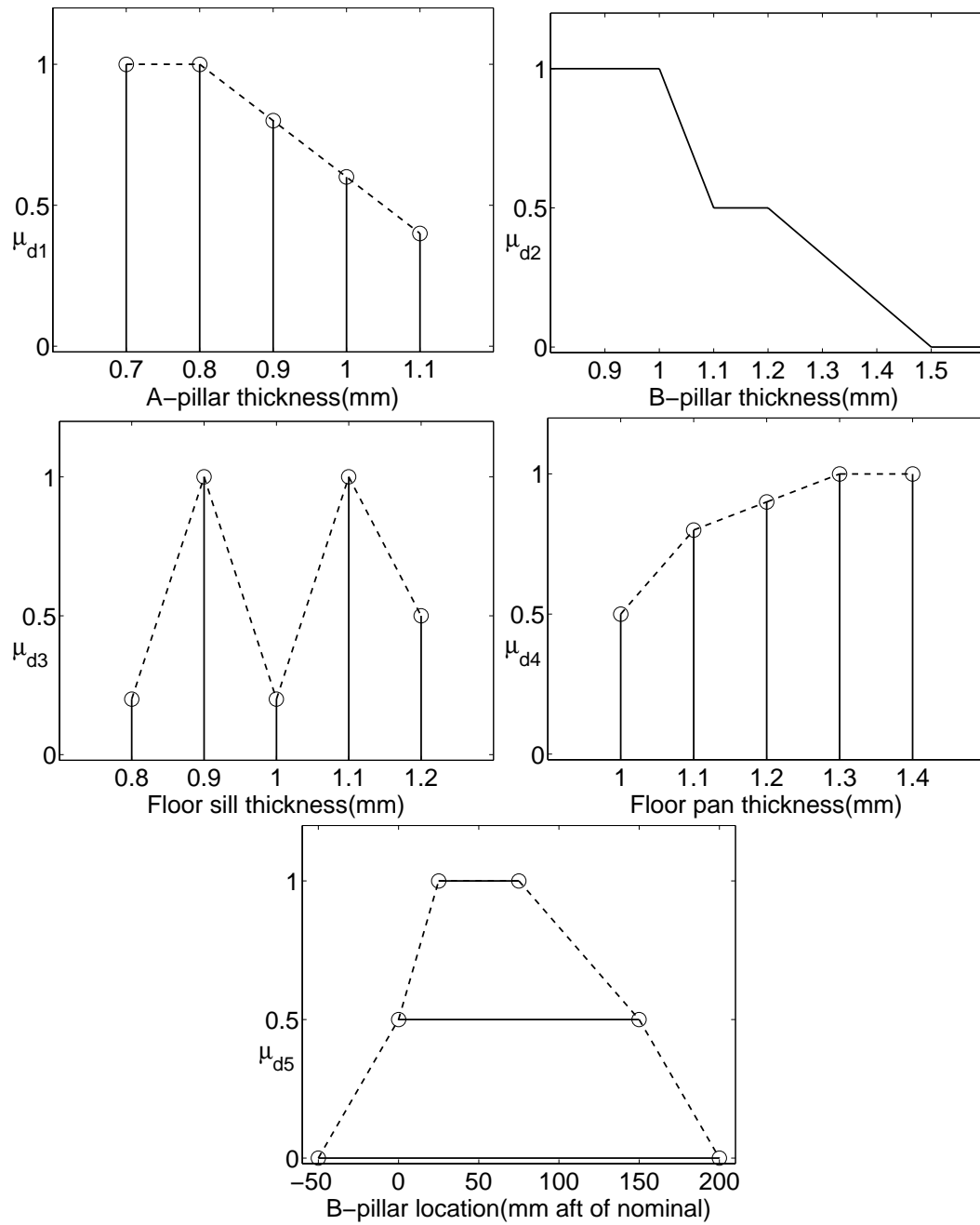


Figure 5.4: Design preferences of the VW model.

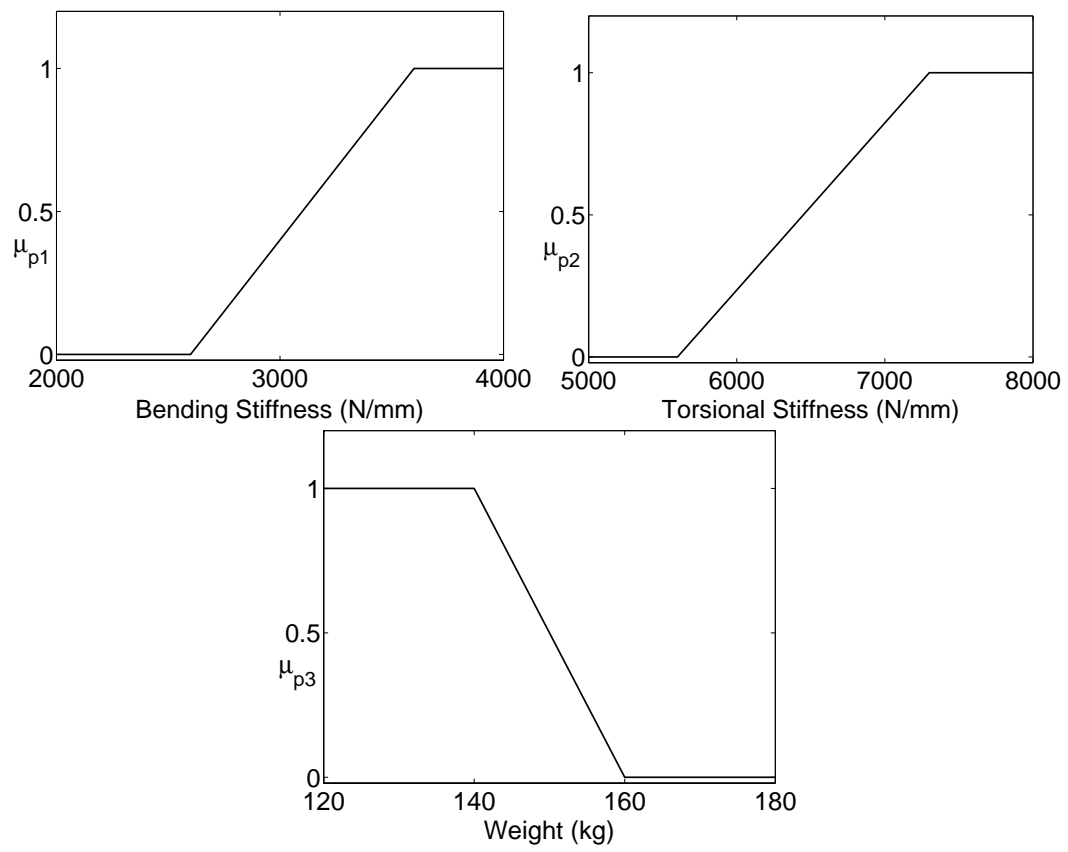


Figure 5.5: Functional requirements of the VW model.

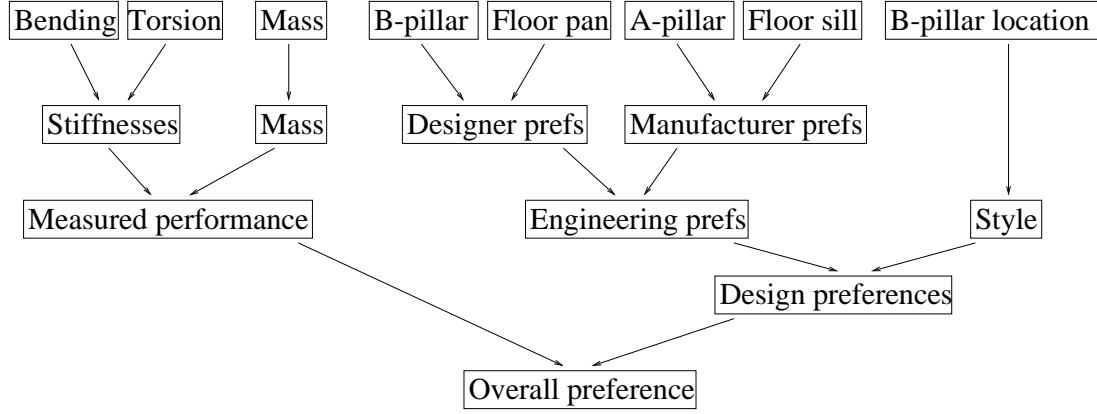


Figure 5.6: Aggregation hierarchy of preferences.

5.2.3 Aggregation of Preferences

After the preferences are defined, the appropriate aggregation of each pair of preferences must be established. The aggregation hierarchy is shown in Figure 5.6. Each aggregation function in the hierarchy is determined as follows:

1. For the functional requirements of bending stiffness and torsional stiffness, there is little compensation in the aggregation of $\mu_{stiffness} = \mathcal{P}_s(\mu_{p1}, \mu_{p2})$ at $\mu = 0.5$:

$$\mathcal{P}_s(0.5, 1.0) \approx \mathcal{P}_s(1, 0.5) \approx \mathcal{P}_s(1.0, 1.0)$$

This means that $\mu_{stiffness} = \min(\mu_{p1}, \mu_{p2})$ is reasonable.

2. For the functional requirement of measured performance, *i.e.*, the combined functional requirement, $\mu_p = \mathcal{P}_s(\mu_{stiffness}, \mu_{p3})$, the indifference points [52] are

$$\mathcal{P}_s(0.3, 1.0) \approx \mathcal{P}_s(1, 0.2) \approx \mathcal{P}_s(0.5, 0.5)$$

The computation shows that $s = -0.02$ and $w = 0.7$. If they are rounded to one decimal place, $s = 0$ which means that $\mu_{measured} = \mathcal{P}_{\Pi}(\mu_{stiffness}, \mu_{p3}; 1, w) = (\mu_{stiffness} \cdot \mu_{p3}^w)^{\frac{1}{1+w}}$.

3. For $\mu_{designer}$, the designer preference aggregated from the preference of the B-pillar thickness (μ_{d2}) and the preference of the floor pan thickness:

$$\mathcal{P}_s(0.4, 1.0) \approx \mathcal{P}_s(1, 0.3) \approx \mathcal{P}_s(0.5, 0.5)$$

These indifference points lead to $s = -1.4$ and $w = 0.6$.

4. The preference of the A-pillar thickness and the floor sill thickness are specified for manufacturability. They are aggregated into the manufacturer preference $\mu_{manufacturer}$. For this aggregation, the indifference points are

$$\mathcal{P}_s(0.4, 1.0) \approx \mathcal{P}_s(1, 0.1) \approx \mathcal{P}_s(0.5, 0.5)$$

The computation shows that $s = -0.2$ and $w = 0.3$.

5. The designer preference and the manufacturer preference will be aggregated into the engineering preference $\mu_{engineering}$. Now the indifference points are found to be:

$$\mathcal{P}_s(0.25, 1.0) \approx \mathcal{P}_s(1, 0.25) \approx \mathcal{P}_s(0.5, 0.5)$$

This shows that the aggregation function is \mathcal{P}_{Π} with weight ratio $w = 1.0$.

6. The next step is to get the combined design preference μ_d by the aggregation of the engineering preference $\mu_{engineering}$ and the preference of the B-pillar location μ_{d_5} which represents the style. This aggregation has the following indifference points:

$$\mathcal{P}_s(0.4, 1.0) \approx \mathcal{P}_s(1, 0.3) \approx \mathcal{P}_s(0.5, 0.5)$$

So $s = -1.4$ and $w = 0.6$ here.

7. Finally, the combined functional requirement and the combined design preference will be aggregated into the overall preference μ_o . This aggregation has the indifference points:

$$\mathcal{P}_s(0.2, 1.0) \approx \mathcal{P}_s(1, 0.3) \approx \mathcal{P}_s(0.5, 0.5)$$

The parameters for the last aggregation function are $s = -0.02 \approx 0.0$ and $w = 1.3$.

Combining all of the above aggregation functions according to the aggregation hierarchy, the final aggregation of the overall preference is

$$\mu_d = \mathcal{P}_{-1.4}(\mathcal{P}_{\Pi}(\mathcal{P}_{-1.4}(d_2, d_4; 1, 0.6), \mathcal{P}_{-0.2}(d_1, d_3; 1, 1); 1, 1), d_5; 0.6, 1) \quad (5.4)$$

Table 5.1: Range of the Design Variables.

Design Variable	d_1	d_2	d_3	d_4	d_5
Minimum Value	0.7	0.9	0.8	1.0	-50
Maximum Value	1.1	1.3	1.2	1.4	150

$$\mu_p = \mathcal{P}_\Pi(\mathcal{P}_{\min}(p_1, p_2; 1, 1), p_3; 1, 0.7)$$

$$\mu_o = \mathcal{P}_\Pi(\mu_d, \mu_p; 1, 1, 3)$$

5.3 Results

Before the solution to the problem is computed, continuous preferences are constructed from the discrete design preferences to facilitate computation. From previous results [51], it is known that the maximum overall preference $\mu_o^* \approx 0.40159$, so the α -levels of interest are set to $\{\varepsilon = 0.01, 0.2, 0.4\}$. If there is no previous result available, the near maximum overall preference can be found at the end of the computation of the first $D_{0.01}^{o\Box}$. The range of each design variable is also restricted to that in the previous work [51] as in Table 5.1. Before the α -cuts are computed, the number of subregions of the range of each design variable is chosen as $\mathcal{S} = 8$, and the number of subregions of the range of each performance variable is chosen as $\mathcal{U} = 2\mathcal{S} = 16$,

First, a Resolution III fractional factorial experimental design is chosen as the design points for the first stage metamodel \vec{f}'_1 . Then the $D_{\alpha_k,1}^{o\Box}$'s are computed for $\alpha_k = 0.01, 0.2$, and 0.4 , and the bounding hypercube of $D_{0.01,1}^{o\Box}$ is found to be:

$$\begin{aligned} 0.7 &\leq d_1 \leq 1.1 \\ 0.9 &\leq d_2 \leq 1.3 \\ 0.8 &\leq d_3 \leq 1.2 \\ 1.0 &\leq d_4 \leq 1.4 \\ -49.95 &\leq d_5 \leq 150 \end{aligned}$$

Another set of 8 design points is sampled within above range, and added to the first 8 design

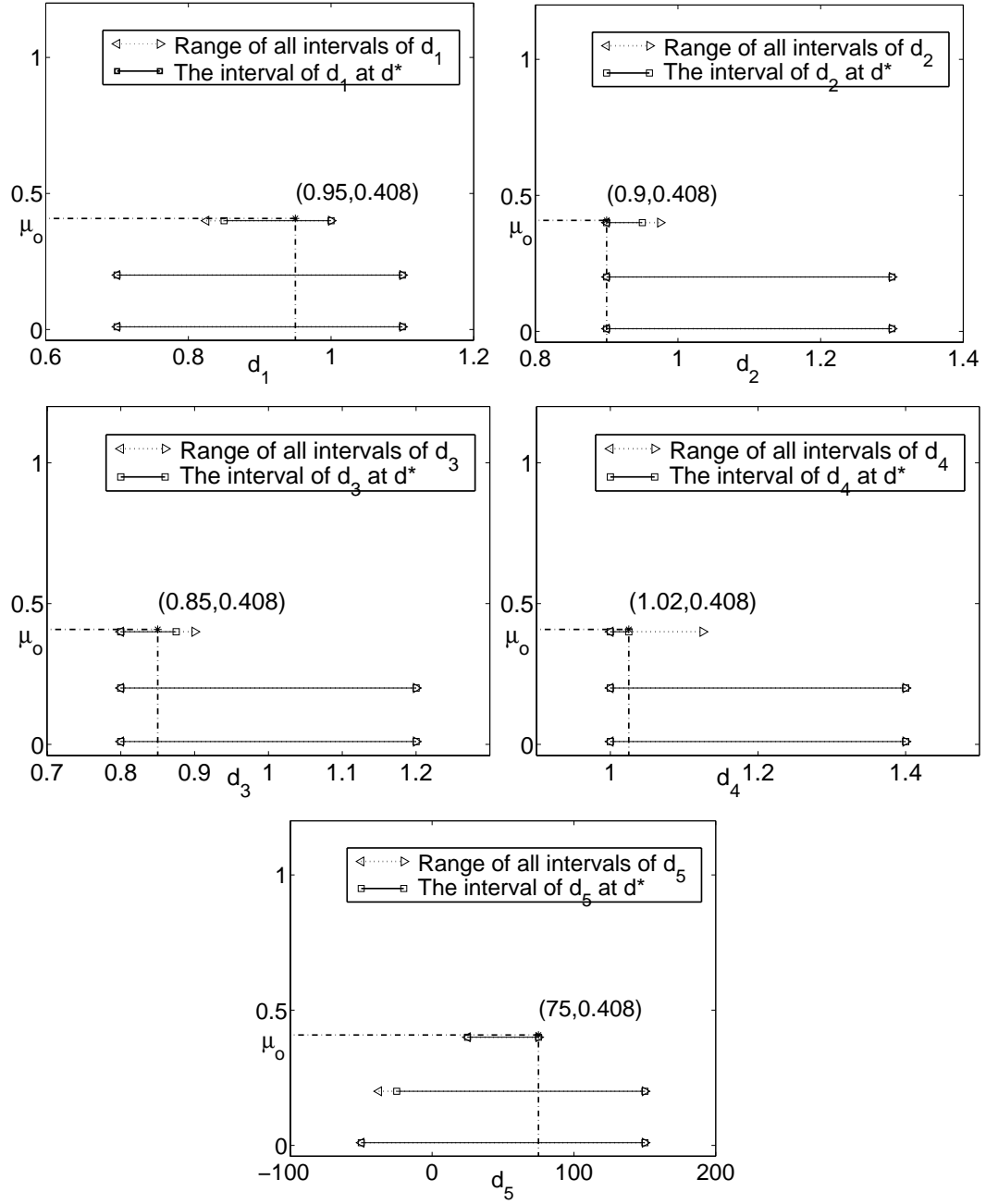


Figure 5.7: The α -cuts of design variables at \vec{d}^* .

points to construct the second metamodel \vec{f}'_2 . Another set of $D_{\alpha_k,2}^{o\Box}$'s is computed with \vec{f}'_2 . Now there are two sets of $D_{\alpha_k}^{o\Box}$'s. The measure $r_{v_d}(\alpha_k, 1, 2)$ for each pair of α -cuts is

$$r_{v_d}(0.01, 1, 2) = 12.77\%$$

$$r_{v_d}(0.2, 1, 2) = 13.61\%$$

$$r_{v_d}(0.4, 1, 2) = 91.13\%$$

These results show that $D_{0.4,2}^{o\Box}$ is the most sensitive to the improvement of the metamodel's accuracy, so the next 8 design points for \vec{f}'_3 should be sampled from the bounding hypercube of $D_{0.4,2}^{o\Box}$ which is

$$0.7 \leq d_1 \leq 1.1$$

$$0.9 \leq d_2 \leq 1.05$$

$$0.8 \leq d_3 \leq 0.925$$

$$1.0 \leq d_4 \leq 1.275$$

$$25 \leq d_5 \leq 100$$

Another 8 design points are sampled within the bounding hypercube of $D_{0.4,2}^{o\Box}$ to build \vec{f}'_3 . After computing $D_{\alpha_k,3}^{o\Box}$'s, the ratio of volume of the change in the α -cuts can be found as:

$$r_{v_d}(0.01, 2, 3) = 4.149\%$$

$$r_{v_d}(0.2, 2, 3) = 5.5\%$$

$$r_{v_d}(0.4, 2, 3) = 75.38\%$$

Compare $r_{v_d}(\alpha_k, 1, 2)$'s and $r_{v_d}(\alpha_k, 2, 3)$'s, it can be noticed that the accuracy of $D_{\alpha_k,2}^{o\Box}$ is improved at all α -levels, especially at 0.01 and 0.2. And the values of $r_{v_d}(\alpha_k, 2, 3)$'s indicate that the 8 design points for \vec{f}'_4 should be sampled within the boundary hypercube of $D_{0.4,3}^{o\Box}$:

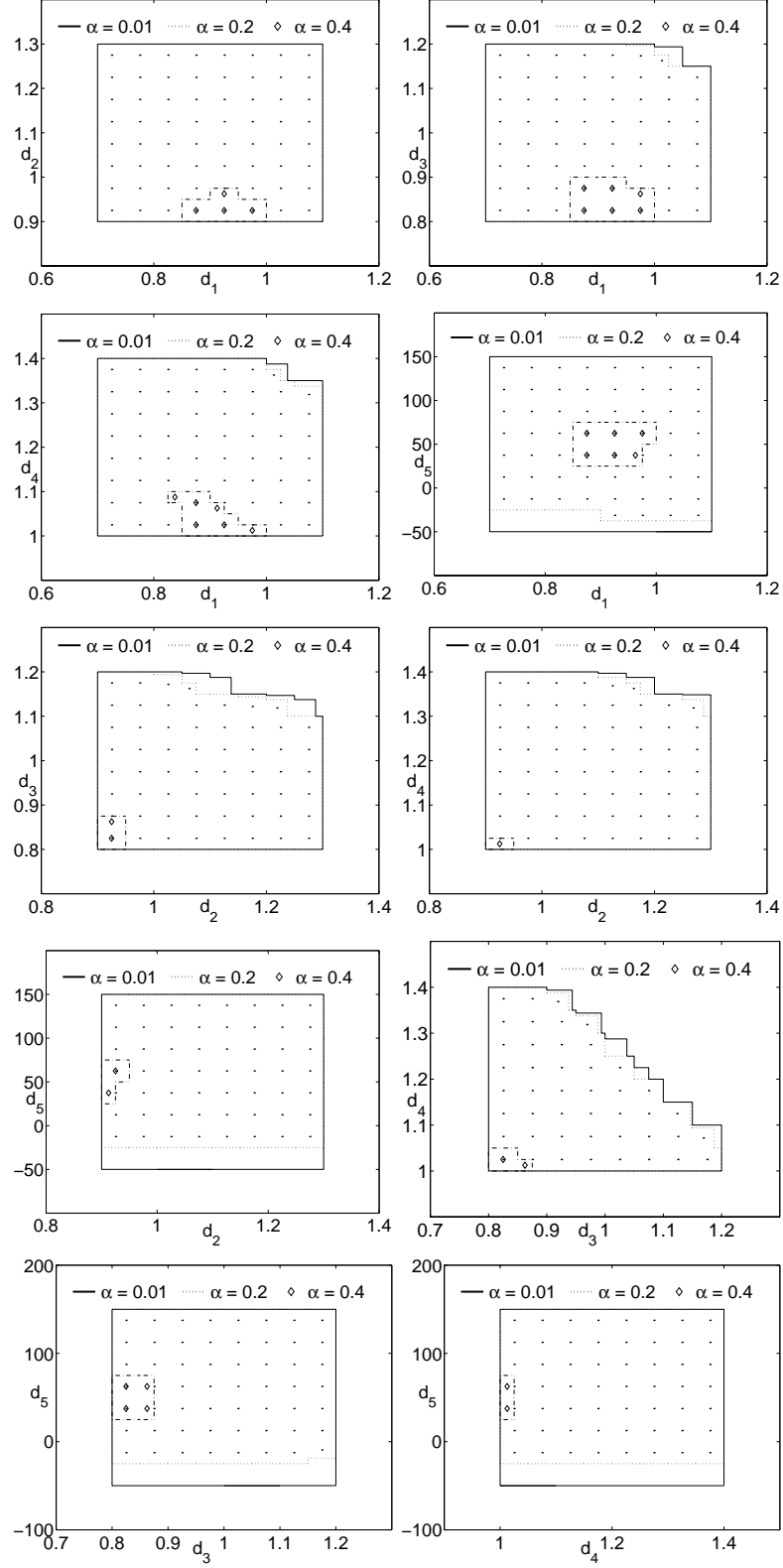


Figure 5.8: The cross sections of $D_{\alpha_k^{o\Box}}$ at \vec{d}^* .

$$\begin{aligned}
0.775 &\leq d_1 \leq 1.05 \\
0.9 &\leq d_1 \leq 1.025 \\
0.8 &\leq d_1 \leq 0.9 \\
1.0 &\leq d_1 \leq 1.175 \\
25 &\leq d_1 \leq 87.5
\end{aligned}$$

Finally, \vec{f}'_4 , the metamodel for the last stage is built, and the final results of $D_{\alpha_k}^{o\Box}$'s and $P_{\alpha_k}^{o\Box}$'s are computed using \vec{f}'_4 . The achievable maximum overall preference $\mu^* = 0.408$. The point in the DVS which can generate μ_* is $\vec{d}^* = (0.95, 0.9, 0.85, 1.02, 75)$. The corresponding point in the PVS with the maximum overall preference is $\vec{p}^* = \vec{f}'_4(\vec{d}^*) = (2,803.09, 5,831.43, 149.33)$. The individual design preferences and the combined design preference at \vec{d}^* are $(0.7, 1.0, 0.6, 0.575, 1.0)$ and 0.8691, respectively. The individual functional requirements and the combined functional requirement at \vec{p}^* are $(0.235, 0.1344, 0.487)$ and 0.2283, respectively.

Here $D_{\alpha_k}^{o\Box}$ is a 5-hypercube, and $P_{\alpha_k}^{o\Box}$ is a 3-hypercube. So for presentation convenience, all α -cuts are shown on the axis of any single variable and as cross sections in the plane of any two variables at \vec{d}^* in the DVS, or at \vec{p}^* in the PVS in Figures 5.7, 5.8, 5.9 and 5.10.

The intervals of the $D_{\alpha_k}^{o\Box}$'s on each design variable at \vec{d}^* are shown in Figure 5.7 with the maximum reachable range of each design variable at \vec{d}^* . The cross sections of $D_{\alpha_k}^{o\Box}$'s are also shown in the plane of any pair of design variables as in Figure 5.8. The cross sections of all three α -cuts are shown in the same figure for comparison.

The intervals of the $P_{\alpha_k}^{o\Box}$ on each performance variable at \vec{p}^* are shown in Figure 5.9 with the maximum reachable range of each design variable at \vec{d}^* . The cross sections of $P_{\alpha_k}^{o\Box}$ are also shown in the plane of any pair of performance variables as in Figure 5.10. Because the shapes of the boundaries $P_{\alpha_k}^{o\Box}$'s are not as simple as those of $D_{\alpha_k}^{o\Box}$'s, each figure only contains one cross section.

5.3.1 Computational Cost

The computation of $D_{\alpha_k}^{o\Box}$'s and $P_{\alpha_k}^{o\Box}$'s at three α -levels $\{0.01, 0.2, 0.4\}$ only requires 32 runs of the finite element model which will take about 32 minutes. The three $D_{\alpha_k}^{o\Box}$'s are computed four times for different metamodels which takes 51 seconds for \vec{f}'_1 , 76 seconds for \vec{f}'_2 , 112 seconds for \vec{f}'_3 , and 135 seconds for \vec{f}'_4 . The computation costs are increasing because of the increases

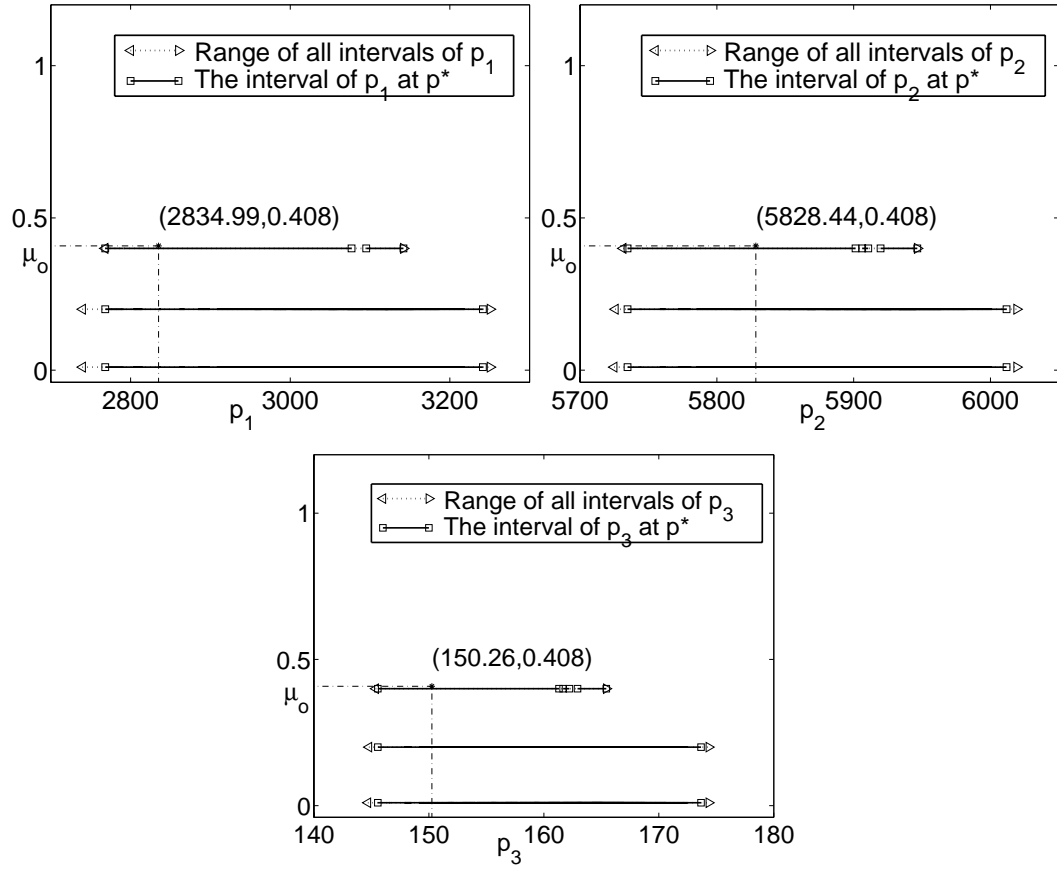


Figure 5.9: The α -cuts of performance variables at \vec{p}^* .

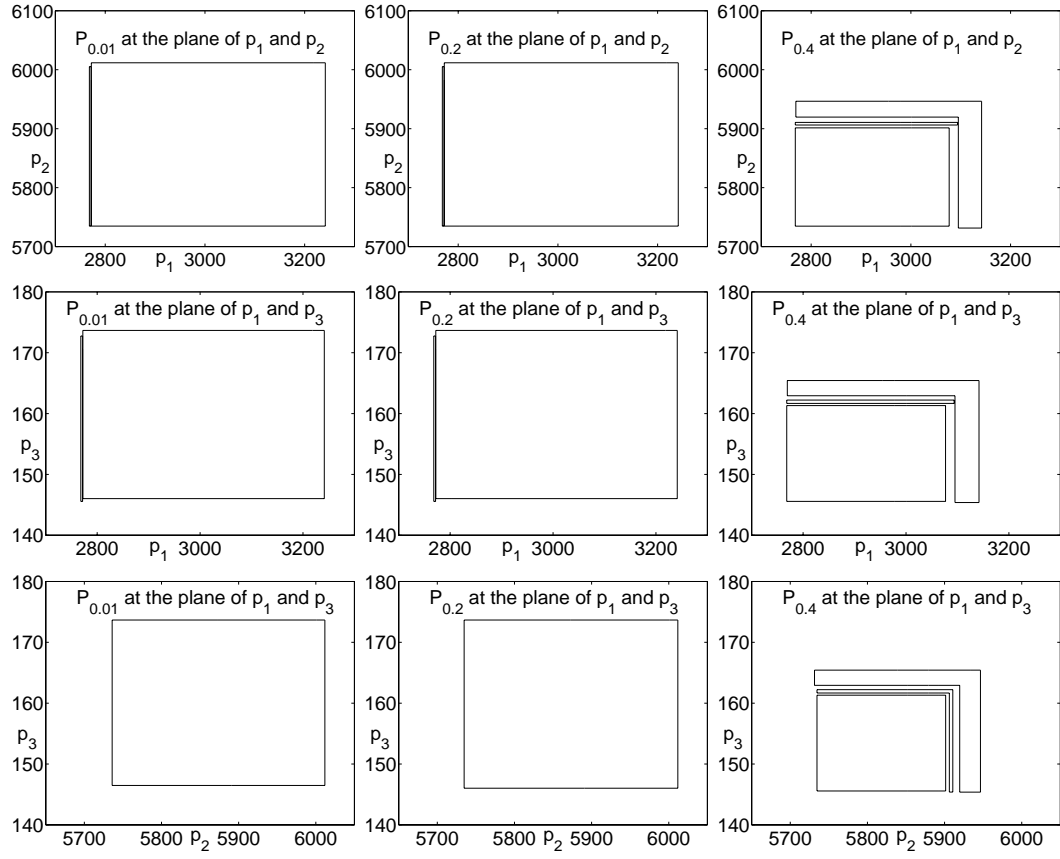


Figure 5.10: The cross sections of $P_{\alpha_k^{\square}}$ at \vec{p}^* .

in the complexity of the metamodels. The total time spent on $D_{\alpha_k}^{o\Box}$ is 374 seconds. The $P_{\alpha_k}^{o\Box}$'s are only computed with the last metamodel \vec{f}_4 , which takes 254 seconds for $\alpha_1 = 0.01$, 214 seconds for $\alpha_2 = 0.2$, and 3 seconds for $\alpha_3 = 0.4$. The computational time of $P_{\alpha_k}^{o\Box}$ is proportional to the number of sub-hypercubes in the DVS at least partially occupied by $D_{\alpha_k}^{o\Box}$ which are 17674 for $\alpha_1 = 0.01$, 16387 for $\alpha_2 = 0.2$, and 27 for $\alpha_3 = 0.4$. The time needed for the r_{v_d} 's is negligible when compared with above costs. The computation cost for the MOI is about 845 seconds (14 minutes). The total time needed to solve this problem is about 46 minutes.

5.4 Discussion

From the resulting $D_{\alpha_k}^{o\Box}$'s shown on one or two design variables, it can be noticed that the differences between the lengths of the intervals or the areas of the cross sections are not significant for α -level at 0.01 and 0.2. This indicates that $\mu_o(\vec{d})$ increases sharply from 0.01 to 0.2, although 0.2 is almost half of the maximum achievable overall preference. $D_{0.4}^{o\Box}$ is quite small compared with the other two, but it is meaningful because $\alpha_3 = 0.4$ is close to $\mu_o^* = 0.408$. If the change between $D_{0.2}^{o\Box}$ and $D_{0.4}^{o\Box}$ is of interest, more α -cuts can be added at α -levels between 0.2 and 0.4.

This problem was also solved in a previous demonstration by using an exhaustive evaluation over all points on a 5^5 grid in the DVS, which takes about 3,000 minutes. The overall preference is computed at the 3,125 points in the DVS. Although the information of μ_o is only available at the 3,125 points, the geometry of $\mu_o(\vec{d})$ can be approximated along the axis of one design variable and in the plane of two design variables. The previous results are $\mu_o^* = 0.40159$ at $\vec{d}^* = (1.0, 0.9, 0.9, 1.0, 50)$, where the design preferences are $(0.6, 1.0, 1.0, 0.5, 1.0)$. The $\vec{p}^* = \vec{f}(\vec{d}^*)$ is $(2,832, 5,836, 147)$, where the functional requirements are $(0.23, 0.14, 0.62)$.

The new results (μ_o^* , \vec{d}^* , and \vec{p}^*) listed in Section 5.3 are close to the previous ones. The biggest difference is between two d_5^* 's. The previous one is 50, and now it is 75. This is because in previous demonstration $\mu_{d_5}(d_5)$ was simplified by connecting $\mu_{d_5}(d_5)$ at $(-50, 0, 50, 100, 150)$, which makes $\mu_{d_5}(75) = 0.75$ although it should be 1.0.

5.5 Summary

In the first section, the ratio of the change in the volume of two α -cuts is proposed as a measure to predict the sensitivity of the α -cut to the change of the metamodel.

The implementation of the M_OI has been applied to a design problem used in a previous demonstration. The specification of preferences and the aggregation strategies are the same as those used previously [51]. The new set-based implementation of the M_OI uses only about 1.5% of the computational time required by the exhaustive evaluation, but generates almost the same results for the maximum overall preference. Besides this result, it creates a set of design alternatives for each α -level of the overall preference. It also provides information about the set of achievable performances with a certain level of overall preference.

Chapter 6

Conclusion

In engineering design, the study of the mapping between the DVS and the PVS is necessary. Usually a simulation model is built for computer analysis software to avoid an expensive physical prototypes. There are many optimization methods which can find the extrema by running the simulation model only a limited number of times. However, in the M_OI , information solely about the extrema is not enough. The whole geometry of the overall preference function, or at least its contours at certain α -levels, is critical to obtain. When the DVS is even moderately high dimensional, the computation to find the geometry of the overall preference function is prohibitively expensive. To implement the M_OI on a real problem, a metamodel, which is “a model of the model” [24], has to be used to reduce the computation cost to a reasonable level.

In previous attempts of using a metamodel, linear polynomial models were used to replace the actual linear or near-linear mapping functions on some performance variables. The simplicity of the linear model is its biggest advantage. It works well for the original Level Interval Algorithm (LIA) in Section 4.1, which requires monotonicity of the mapping function. However, for any extension of the LIA which has the ability to generate more accurate results, the linear model can not satisfy the accuracy requirement. Moreover, the linear model is not flexible enough. Increasing the number of design points is not helpful to increase the accuracy of the linear model.

A nonlinear interpolation model is introduced into the M_OI in Section 3.2. This interpolation model can be considered as the combination of a linear approximation built by the generalized least-square regression method, and a nonlinear compensation for the approximation error based on the assumption that the error is an instance of a Gaussian process. Among the several base functions listed in Section 3.4, the simplest linear polynomials are chosen in Section 3.5. The reasons for this choice are as follows: the chosen base function preserve the simplicity of the linear polyno-

mial approximation model, they follow the “main effects principle,” and the flexibility requirement for the approximation is satisfied by the nonlinear part of the interpolation model. For the linear polynomial, which contains mostly the main effect, the Resolution III fractional factorial design of experiment is the best candidate among all the experimental design methods in Section 3.3.

The design points chosen by the experimental design are decided by the type of base function. If more design points are added afterwards, the accuracy of the linear approximation will not change much, but the errors in the region near the response of any design point will be reduced. It would be helpful to find a sampling criterion for the design points after the original experimental design which can increase the accuracy of the metamodel significantly. Many criteria are discussed in Section 3.6. Each criterion is based on some assumptions. The optimal set of design points under any sampling criterion is the set of design points, which maximizes or minimizes the objective function of the sampling criterion. In engineering design, the type of mapping function is unconstrained. It is not practical, if not impossible, to prove whether the assumption of any sampling criterion is violated or not, so two testing examples are used to compare these sampling criteria, and the empirical root-mean-square error is used as a measure of the accuracy of the metamodel. The test results show that there is no evidence to back the conclusion that any sampling criterion is better than any other with respect to improving the accuracy of the metamodel.

Two tests of the improvement in accuracy of the metamodels were also carried out at the same time. The results suggest that the improvement of accuracy is noticeable. The indifference between sampling criteria is not encouraging but understandable. Some researchers get similar results in comparisons between many sampling criteria [7]. The equivalent density of 8 points for a five-dimensional DVS is equal to $8^{1/5} \approx 1.5$ points for a one-dimensional DVS. To double the equivalent density, 3 points are needed for the 1-dimensional DVS, but $8 \cdot 2^5 = 256$ points will be needed for the 5-dimensional DVS. With 32 points in the 5-dimensional DVS, the equivalent density is $32^{1/5} = 2$, which is only about 1.3 times the equivalent density of 8 points although the number of design points is already 4 times as much.

When the dimensionality of the DVS increases, the improvement of the equivalent density will deteriorate compared with the absolute number of design points. When the validity of the assumptions of the sampling criteria is unknown, the equivalent density of the design points will dominate the increase in the performance of the metamodel. Because the increase in accuracy in the region near added design points can be induced from the structure of the interpolation model, the sampling range is more crucial than the sampling criterion for newly added design points.

After the metamodel is constructed, it can be used to induce preference in one space from that in the other space by using the extension principle. The LIA is an efficient implementation of the extension principle. It requires monotonicity of the mapping function to generate the correct answer. It avoids solving a nonlinear programming problem by sacrificing accuracy. The assumption about the aggregation function of the preferences and the assumption about the independence between the mapping functions limits the area of its application if reasonable accuracy is required. The new extension of the original LIA proposed in Section 4.3 relaxes these assumptions, and reduces the errors if the relaxed assumptions are violated.

The target of the M_OI is to find the set of designs that exhibit a certain level of overall preference for their performances. The LIA is a good implementation of the extension principle for this purpose because it approximates the overall preference by α -cuts. But when, and how, to use the LIA still remains a question because the mapping function from the PVS onto the DVS is usually unknown or does not even exist. In the previous computational implementation of the M_OI , which consists of the forward calculation and the backward path, the LIA is used twice. First the combined design preference in the PVS $\mu_d(\vec{p})$ is induced from the combined design preference in the DVS $\mu_d(\vec{d})$ in the forward calculation. Then the LIA is used to induce the overall preference in the DVS $\mu_o(\vec{d})$ from the overall preference in the PVS $\mu_o(\vec{p})$ in the backward path. In the forward calculation, any aggregation function other than $\min(\mu_1, \mu_2)$ will create many intermediate α -levels of the combined design preference if the design preferences are divided at α -levels besides $0 < \varepsilon \ll 1.0$ and 1.0 . For a n -dimensional DVS, if each design preference is divided into three α -cuts at $\{\varepsilon, \alpha, 1.0\}$ where $\varepsilon < \alpha < 1.0$, there will be n more new α -levels between α and 1.0 in the worst case. If these new α -cuts are ignored, some information will be discarded. If they are included in later computations, $n + 3$ α -levels may be too many for the designer's discretion. In the backward path, the LIA requires the inverse of the performance functions, although the solution can be found without them for problems which satisfy some special requirements [27].

A new method to compute the overall preference is presented and discussed in Section 4.4. Because the discretization is only applied to the overall preference at the last step of the computation after any aggregation in the new method, no new α -levels will be created. The LIA will only be used once to induce the overall preference in the PVS from that in the DVS, in which only the forward mapping function from the DVS to the PVS is needed. This is because the forward mapping function $\vec{p} = \vec{f}(\vec{d})$, which is used to compute the performances of a design, is used twice. The first use of \vec{f} is in the computation of the combined functional requirement in the DVS $\mu_p(\vec{d}) = \mu_p(\vec{f}(\vec{d}))$,

and \vec{f} is also used to induce the α -cut of the overall preference in the PVS P_α^o from the α -cut of the overall preference in the DVS D_α^o . The LIA is a discrete approximate implementation of the extension principle. It only needs the overall preference $\mu_o(\vec{d})$ at some individual points in the DVS to compute P_α^o . So, $\mu_p(\vec{d})$ is needed only at some individual points in the DVS. \vec{f} is a N to 1 mapping from the DVS onto the PVS where $N \geq 1$. However, for the set which contains any individual point in the DVS, it becomes a 1 to 1 mapping, and hence has an inverse. The extension principle is simplified to $\mu_p(\vec{d}) = \mu_p(\vec{p}) = \mu_p(\vec{f}(\vec{d}))$ for the 1 to 1 mapping at any individual point in the DVS. Hence the use of \vec{f}^{-1} is avoided.

Now the α -cuts of the overall preference can be computed more accurately without the inverse of the forward mapping function. If the metamodels are used in the computation, the boundary of each α -cut has different levels of sensitivity to the increase in accuracy of the metamodel. It is more cost-effective to add new design points for the metamodel in the region within the most metamodel-sensitive α -cut but outside the α -cut at the next higher α -level, because the study in Chapter 3 shows that where to sample the new design points is more important than how to sample them when the equivalent density of design points is low. The difference in the volume of two α -cuts at the same α -level, but computed from different metamodels, is proposed in Section 5.1.1 as the measure of the sensitivity of the α -cut at a certain α -level to the change in the metamodel. The ratio of the difference in volume to the original volume is preferred to compare the sensitivity of α -cuts at different α -levels.

Finally, the whole computation process of the M₀I is formed from the preceding discussion in Section 5.1.2 and is applied to the design problem described in Section 5.2 which was used to demonstrate the M₀I. The results listed in Section 5.3 include α -cuts at $\{\varepsilon = 0.01, 0.2, 0.4\}$, and information about the maximum overall preference. Compared with previous results, the new computation method produces almost the same maximum overall preference, but finds the design which has the maximum overall preference within only a small portion, about 1.5%, of the time used by the previous demonstration.

There are three parameters of the computation which decide the accuracy and the cost of the M₀I:

1. The number of design points used to construct the metamodel.
2. The number of subregions of the range of each design variable.
3. The number of subregions of the range of each performance variable.

There are mainly two contributions to the computational cost: the cost to evaluate the mapping functions between the DVS and the PVS, which is decided by the first of the listed parameters; and the cost to compute the α -cuts of the overall preference in the DVS and the PVS, which is decided by the last two parameters. The accuracy of the resulting α -cuts is also affected by two factors: the accuracy of the metamodel of the actual mapping function, which is controlled by the first parameter; and the accuracy of the LIA, which is controlled by the last two parameters. A larger value of any of these three parameters produces a higher computation cost but more accurate results.

The questions posed in Chapter 1 can now be answered:

1. A more accurate approximation than the linear approximation is preferred.
2. A nonlinear metamodel is not only feasible, but is necessary for the new extension of the LIA.
3. The accuracy of the mapping of the α -cut is improved by dividing the range of each variable into smaller regions. Besides, the α -cuts in the DVS and the PVS can be represented in more detail than just using a hypercube.
4. With the metamodel and the new extension of the LIA, the designer has more control over the trade-off between computation cost and accuracy.

Bibliography

- [1] J. Aczél. *Lectures on Functional Equations and their Applications*. Academic Press, New York, 1966.
- [2] S. Baas and H. Kwakernaak. Rating and ranking of multiple-aspect alternatives using fuzzy sets. *Automatica*, 13:47–48, 1977.
- [3] R. A. Bates, R. J. Buck, E. Riccomagno, and H. P. Wynn. Experimental design and observation for large systems. *Journal of the Royal Statistical Society Series B- Methodological*, 58(1):77–94, 1996.
- [4] R. E. Bellman and L. A. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 17(4):B–141–163, December 1970.
- [5] George E. P. Box and D. W. Behnken. Some new three-level designs for the study of quantitative variables. *Technometrics*, 2:455–477, 1960.
- [6] George E. P. Box, William G. Hunter, and J. Stuart Hunter. *Statistics for Experimenters*. Wiley, New York, NY, 1978.
- [7] Wei Chen, Ruichen Jin, and Agus Sudjianto. On sequential sampling for global metamodeling in engineering design. In *Proceedings of DETC02: 2002 ASME Design Engineering Technical Conference*, 2002.
- [8] B. Cheng and D. M. Titterington. Neural networks, a review from a statistical perspective. *Statistical Science*, 9(1):2–30, 1994.
- [9] D. R. Cox and N. Reid. *The theory of the design of experiments*. Chapman and Hall/CRC, Boca Raton, 2000.
- [10] Selden B. Crary, Peter Cousseau, and Dabid Armstrong. Optimal design of computer experiments for metamodel generation using i-opt (tm). *CMES-Computer Modeling in Engineering and Sciences*, 1(1):127–139, 2000.
- [11] William J. Diamond. *Practical experiment designs for engineers and scientists*. John Wiley and Sons, Inc., New York, New York, 2001.
- [12] 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.

- [13] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [14] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Information Sciences*, 36:85–121, 1985.
- [15] Richard O. Duda and Peter E. Hart. *Pattern classification and scene analysis*. Wiley, New York, NY, 1973.
- [16] John Anderson et al. *Machine learning : an artificial intelligence approach*. M. Kaufmann, Los Altos, California, 1983.
- [17] Leonard Bolc et al. *Computational models of learning*. Springer-Verlag, New York, NY, 1987.
- [18] Jerome H. Friedman. Fitting functions to noisy data in high dimensions. In *Computing Science and Statistics*, pages 13–43. American Statistical Association, April 1988. Proceedings of the 20th Symposium on the Interface.
- [19] Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–141, 1991.
- [20] 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, NY, 1974. Academic Press.
- [21] Maurice F. Holmes. Machine dynamics, The need for greater productivity. In K. N. Reid, editor, *Research Needs in Mechanical Systems*, pages 140–159, New York, NY, 1984. ASME.
- [22] M.E. Johnson, L.M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990.
- [23] J. Kiefer and J. Wolfowitz. Optimum design in regression problems. *Annals of Mathematical Statistics*, 30:271–294, 1959.
- [24] J. P. C. Kleijnen. *Statistical tools for simulation practitioners*. Marcel Dekker, New York, 1987.
- [25] P. Langley and H.A. Simon. Applications of machine learning and rule induction. *Communications of the ACM*, 38(11):55–64, 1995.
- [26] William S. Law. *Evaluating Imprecision in Engineering Design*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, June 1996.
- [27] 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.

- [28] 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.
- [29] 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.
- [30] 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.
- [31] William S. Law and Erik K. Antonsson. Multi-dimensional Mapping of Design Imprecision. In *8th International Conference on Design Theory and Methodology*. ASME, August 1996.
- [32] Jeffrey K. Liker, Durward K. Sobek, Allen C. Ward, and John J. Cristiano. Involving suppliers in product development in the United States and Japan: Evidence for set-based concurrent engineering. *IEEE Transactions on Engineering Management*, 43(2):165–178, 1996.
- [33] D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, (27):986–1005, 1956.
- [34] J. M. Lucas. How to achieve a robust process using response-surface methodology. *Journal of Quality Technology*, 26(4):248–260, 1994.
- [35] M. D. McKay, R. J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, May 1979.
- [36] T. Mitchell, J. Sacks, and D. Ylvisaker. Asymptotic bayes criteria for nonparametric response-surface design. *Annals of Statistics*, 22(2):634–651, 1994.
- [37] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, New York, NY, 1991.
- [38] Kevin N. Otto. *A Formal Representational Theory for Engineering Design*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, June 1992.
- [39] Kevin N. Otto and Erik K. Antonsson. Some α -cut Concepts. Engineering Design Research Laboratory Report EDRL-TR 91c, California Institute of Technology, 1991.
- [40] Kevin N. Otto and Erik K. Antonsson. Trade-Off Strategies in Engineering Design. *Research in Engineering Design*, 3(2):87–104, 1991.
- [41] 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.

- [42] Kevin N. Otto and Erik K. Antonsson. Tuning Parameters in Engineering Design. *ASME Journal of Mechanical Design*, 115(1):14–19, March 1993.
- [43] Kevin N. Otto, Andrew D. Lewis, and Erik K. Antonsson. Approximating α -cuts with the Vertex Method. *Fuzzy Sets and Systems*, 55(1):43–50, April 1993.
- [44] 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.
- [45] Sudhendu Rai. Optimal experimental setpoint determination in systems with multiple output responses. In *Proceedings of DETC98: 1998 ASME Design Engineering Technical Conference*. ASME, September 1998. Paper Number DETC98/DTM-5678.
- [46] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wym. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, February 1989.
- [47] K. J. Schmucker. *Fuzzy Sets, Natural Language Computations, and Risk Analysis*. Computer Science Press, Inc., Rockville, Maryland, 1984.
- [48] Michael J. Scott. *Formalizing Negotiation in Engineering Design*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, June 1999.
- [49] Michael J. Scott and Erik K. Antonsson. A Comparison of Design Evaluation Techniques. September 1998.
- [50] Michael J. Scott and Erik K. Antonsson. Aggregation Functions for Engineering Design Trade-offs. *Fuzzy Sets and Systems*, 99(3):253–264, 1998.
- [51] Michael J. Scott and Erik K. Antonsson. Preliminary Vehicle Structure Design: An Industrial Application of Imprecision in Engineering Design. In *10th International Conference on Design Theory and Methodology*. ASME, September 1998.
- [52] Michael J. Scott and Erik K. Antonsson. Using Indifference Points in Engineering Decisions. In *11th International Conference on Design Theory and Methodology*. ASME, September 2000.
- [53] P. Sebastiani and H. P. Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society Series B- Statistical Methodology*, 62(1):145–157, 2000.
- [54] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *J. Applied Statistics*, 14:165–170, 1987.
- [55] Durward K. Sobek, Allen C. Ward, and Jeffrey K. Liker. Toyota’s principles of set-based concurrent engineering. *Sloan Management Review*, 40(2):67–+, 1999.
- [56] L. P. Sullivan. Quality function deployment. *Quality Progress*, 19(6):39–50, June 1986.
- [57] Karl T. Ulrich and Scott A. Pearson. Does product design really determine 80% of manufacturing cost? Working Paper MSA 3601-93, MIT, Sloan School of Management, Cambridge, MA, August 1993.

- [58] Allen C. Ward, Jeffrey K. Liker, John J. Cristiano, and Durward K. Sobek. The second Toyota paradox: How delaying decisions can make better cars faster. *Sloan Management Review*, 36(3):43–61, Spring 1995.
- [59] 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.
- [60] D. E. Whitney. Manufacturing by design. *Harvard Business Review*, 66(4):83–91, July 1988.
- [61] Kristin L. Wood. *A Method for Representing and Manipulating Uncertainties in Preliminary Engineering Design*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1989.
- [62] 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.
- [63] 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.
- [64] 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.
- [65] R. Yager. Fuzzy decision making including unequal objectives. *Fuzzy Sets and Systems*, 1:87–95, 1978.
- [66] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

Index

- \mathcal{P}_s , *see* weighted root-mean-power
- aggregation function, 8
- aggregation hierarchy, 13
- aliases, 21
- α -cut, 70
- $D_{\alpha_k}^d$, 70
- $D_{\alpha_k}^{d\Box}(\hat{s})$, 83
- $D_{\alpha_k}^{d\Box}(S)$, 83
- $P_{\alpha_k}^{d\sharp}$, 80
- $P_{\alpha_k}^d$, 78
- $P_{\alpha_k}^{d\Box}(S, \mathcal{U})$, 84
- Annihilation, 9
 - weighted, 11
- anomalies of the LIA, 73
- Backward Path, 87
- Bayesian I-Optimal design, 40
- body-in-white*, 101
- Box-Behnken design, 21
- CCD, *see* central composite design
- central composite design, 21
- combined design preference, 68
- combined functional requirement, 68
- Commutativity, 8
 - weighted, 10
- computation of $D_{\alpha_k}^o$ and $P_{\alpha_k}^o$ without \vec{f}^{-1} , 88
- computer experiment, 18
- Continuity, 8
 - weighted, 10
- covariance structure, 18
- design appropriate, 9
- design points, 19
- design preference, 7
- design variable space, 6
- design variables, 6
- DV, *see* design variables
- DVS, *see* design variable space
- equivalent density, 118
- ERMSE, 28
- extension of the LIA, 79
- extension principle, 69
- factorial design, 20
- 2^k factorial design, 20
- Forward Calculation, 87
- 2^{k-p} fractional factorial design, 21
- fully compensating, 9
- functional requirement, 7
- G-optimal, 39
- Gaussian process, 18
- geometric weighted mean, 9
- Idempotency, 9
 - weighted, 11
- IDT, 2
- implementation of the M_0I , 100
- imprecision, 1
- indifference points, 13
- interpolation model, 19
- I_z -Optimal, 40
- Latin Hypercube, 20
- Level Interval Algorithm, 70
- LIA *see* Level Interval Algorithm 70
- $\mu_{d_i, \vec{r}}(d_i)$, 82
- main effects principle, 22
- mapping function, 7
- maximin distance design, 39
- maximum entropy sampling criterion, 38
- metamodel, 15
- Method of Imprecision, 1, 6
- Monotonicity, 8

- weighted, 10
- $\mu_d(\vec{d})$, 68
- $\mu_d(\vec{p})$, 69
- $\mu_o(\vec{d})$, 8
- $\mu_o(\vec{p})$, 8
- $\mu_p(\vec{p})$, 68
- non-compensating, 9
- overall preference, 8
- partially compensating, 12
- Parzen Windows method, 65
- performance variable, 6
- performance variable space, 7
- polyhedron approximation, 80
- PV, *see* performance variables
- PVS, *see* performance
 - variable space
- randomization test, 43
- $r_{v_d}(\alpha_k, l, l + 1)$, 99
- Resolution
 - III, 21
 - IV, 21
 - V, 21
- Self-normalization, 11
- set-based, 2
- significance level
 - F , 42, 45
 - t , 42, 45
- sub-hypercube
 - \mathcal{X}_{i,r_i} , 82
 - \mathcal{Y}_t^* , 83
- super-compensating, 12
- systematic departure, 18
- test points, 24
- Vertex Method, 70
- $V(D_{\alpha_k,l+1}^{o\Box} - D_{\alpha_k,l}^{o\Box})$, 99
- weighted means, 11
- weighted root-mean-power,
 - 12
- weights, 10