A Formal Representational Theory for Engineering Design

Thesis by
Kevin N. Otto

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

California Institute of Technology
Pasadena, California

1992

(Defended May 1, 1992)

Acknowledgments

This document and the work it represents was impossible without the support of my wife Ginger. Often one needs non-technical advice to make clear what one is contemplating. Also one always needs a financial supporter.

My thesis advisor Erik Antonsson helped focus many of my thoughts. In addition to providing me with technical assistance, he as well provided instruction on the process of conducting academic research, the communication of ideas both orally and written, and the approach to a developing field.

I also owe much to my colleagues in the Engineering and Applied Science Division at Caltech. Their comments and advice maintained my comprehension and rigor. Andrew Lewis in particular provided me with invaluable support. Many of the technical proofs were impossible without him.

This material and the work it represented were made possible, in part, by a fellowship from the AT&T-Bell Laboratories Ph.D. scholar program, sponsored by the AT&T foundation. Also, the National Science Foundation provided funding under a Presidential Young Investigator Award, Grant No. DMC-8552695, through my thesis advisor Dr. Erik Antonsson. Any opinions, findings, conclusions or recommendations expressed in this publication are my own and do not necessarily reflect the views of the sponsors.

A Formal Representational Theory for Engineering Design

by

Kevin N. Otto

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

Abstract

In the design of engineered artifacts, it is hypothesized that computations must be performed. Informal specifications must be converted into formal functional requirements and informal descriptions must be converted into formal parameterizations, so that performance can be computed. Such performance evaluations are developed for general set based mappings. The development includes, for example, functional relations, differential equations, simple experimentation, and even subjective questioning. When this level of formalization is complete, the design is not determined; it is only parameterized. A designer must specify levels of the various performances desired, and how the various performances should be simultaneously considered in an overall determination. An axiomatically based methodology is presented to formalize such decisions. Each decision variable is equipped with a preference specification, whose determination is made from techniques similar to utility theory. A design strategy for resolving these different aspects of a design is developed to produce an overall rating. For example, a designer may rate a design by the worst case performance. Alternatively, a designer may rate a design by a compensation among the goals. In addition to decision representation, other parameters in a formalization reflect phenomena which the designer cannot control. A methodology for accommodating confounding noise influences is developed. Random measurement noise is represented, as well as the possibility of other decision makers in a design process. Convolutions of these methods are developed. For example, designer decision-making is developed for a decision which must be made in light of random errors (such as manufacturing). Designer decision-making is further developed for the case when a manufacturing engineer can subsequently tune a design (a possibilistic uncertainty) to eliminate such random error effects. Ensuring against failure is also discussed, with respect to the measured noise. Given this development, a methodology is constructed in which a designer can incompletely specify performance requirements on a design. The incomplete specification can be induced across the design, to determine any restrictions imposed on the portions of the design where no specifications have been made. Thus, an iterative design process of specification, calculation, observation, and re-specification is given formal foundation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A typical engineering design process involves the satisfaction of needs through the development of a product. There have been several historical models of engineering design as a process. Asimow [9] or Pugh [71, 126] describe a transition model of preliminary design through detailed design through manufacturing. Ullman [179] also presents design as refinement of descriptions. Suh [166] describes design as a transition from functional descriptions to form descriptions. Pahl and Beitz [121] describe a model of stages with function to form transformations. Adams [2] considers design as the steps of synthesis, analysis, and decision. Yoshikawa [174, 197, 198, 199] describes design as mappings between sets.

These models of design processes are insufficient as a representational theory of engineering design. They fail to distinguish between concepts which are formal model abstractions (and can be computed with), and those which are not. The desires, intentions, and beliefs of designers, customers, and those who interact in a design process cannot be absolutely modeled, they can only be formally represented. Design occurs to satisfy "needs" and "needs" are informal concepts, which can only be formally represented. All computational models of design processes operate not with true intentions and desires, but with formal reflections of the true intentions, which may or may not be "accurate."

Thus, this thesis clarifies the need for converting informal descriptions into formal models. Others also observe this need. The central theme of QFD [4, 69] is "voice of the customer," meaning one must determine what the needs of the customer are, and perform all design activities to maximize customer satisfaction. Also, in the architectural design domain, the design phase of programming [122] is, similarly, the phase where the architect translates the customer's desires, as described by the customer and interpreted by the architect, into more formal design model requirements. Other design models also observe the requirement of determining customer needs [9, 90, 121]. The difference between informal descriptions and formal reflections which can be measured, however, is not emphasized.

This thesis discusses formal models in engineering design, on which computation can occur. A paradigm is theorized in which informal expressions of need are interpreted by a designer, and converted into formal functional requirements which can be computed, given a formal design model. This is extended to include the formal design model itself as a reflection of a designer's informal initial interpretation of a design concept. Thus a theory is developed depicting a design process starting with informal intentions and interpretations of concepts and needs, which are formalized into a computational model.

Having a formalization of a design, however, is not the same as having a solution to the formalization. Given parameterized design options and parameterized needs, the particular option to use must be determined. Conflicting goals must be resolved. Parameters the designer cannot control (noise) must be considered. Thus interaction occurs between informal intentions and formal models.

1.1 Goals and Motivation

This thesis attempts to provide an axiomatic foundation for designer interaction with a formal model. Given conflicting goals within a formal model, restrictions are developed to ensure particular resolution methods are appropriate. These restrictions can then be informally considered by the designer, and adopted or rejected, thereby adopting or rejecting the goal resolving methods. Given uncontrollable parameters within a formal model, restrictions are developed to ensure particular methods for incorporating noise are appropriate. These restrictions can then as well be informally considered by the designer, and adopted or rejected, thereby adopting or rejecting the noise model.

The development of such work is important for the development of design theory. If it can be shown that there are aspects of a design process which are necessarily informal, erroneous attempts at making these informal aspects formal need not occur. Also, particular formalizations used in practice can be seen in a new light, under the axiomatic foundation developed. Restrictions that existing methods conform to can be scrutinized.

The aim of the thesis is a complete axiomatic foundation of a design process. Every formalization step used in a design process should have an associated informal justification, so that a designer can consider the formalization step. Wherever possible, assumptions have been made explicit. Assumptions, though, are usually clear only in hindsight. Within this thesis, it is more than likely that many assumptions remain unexpressed. Nonetheless, the development remains of value, in that making a process partially explicit is better than not formally understanding it at all.

1.2 Approach

It is proposed that a defining aspect of engineering design is the translation of informal intentions, beliefs, and interpretations into formal descriptions on which computations can occur. Engineering design, as desired to be modeled by this thesis, requires computations on formal models. It is proposed that a formal model can be characterized by set theory [17, 65, 167], distinguishing formal models (sets) from informal characterizations, and computation (mathematical operations) from informal reasoning.

After formalizing a design, informal characterizations can continue to enter the design. Design options are formally parameterized, but the selection of a final design is based on informal needs. It is proposed to formally and incrementally restrict the selection process, while informally justifying each formal restriction. With sufficient number, the restrictions specify unique formal operators which can be used.

The same approach is taken to model uncontrollable parameters. Typically in engineering design, uncontrollable parameters are formally parameterized, but their actions are accounted for by an informal decision. Again, it is proposed to restrict the accounting process, and informally justify the restrictions.

This approach will therefore provide a methodology which allows a designer to specify *what* should be determined in a design model. The analysis has not developed a methodology for *how* a designer should go about this, in a process sense. To consider this question, the approach is taken that a designer can *partially* specify a design, and compute with this partial specification. Doing so will allow a designer to determine the performance achievable, given the partial specification. Such an approach is developed, using the formalization framework of the previous discussion.

1.3 Organization of Thesis

The thesis presents developments in a constructive form. Each chapter requires the preceding chapters for its developments.

Chapter 2 presents a hypothesized description of formal engineering design. The distinction is made between informal interpretations and intentions, and formal computational reflections of these concepts. The performance improvement principle is introduced. A formal/informal design model interaction paradigm is presented. A formal engineering model is defined using basic set theory. Justifications are given for each development based on informal arguments and industrial practices.

Chapter 3 presents the informal concepts of different uncertainty forms. Imprecision is defined as designer uncertainty in choice. This is developed complete with trade-off strategies among different goals. Probability is defined as random variations. Possibility is defined as uncertainty in capacity. Also, necessity is defined as uncertainty which must be ensured. Again, justifications are given for each development based on informal arguments and industrial practices.

Chapter 4 develops formal techniques for solving design problems which exhibit only imprecision. Section 4.2 develops the formal design and performance parameter spaces, and the specification of preference. Weighting functions are developed. Section 4.4 defines the "solution" to a design problem, based on the formalization of the previous sections and chapters. Section 4.5 formally develops various preference metrics which could be used to define performance. Section 4.5.3 discusses other metrics which have been used, but do not conform with the development of Section 4.5. Section 4.6 presents examples using the development of Chapter 4. Section 4.7 discusses additional restrictions which reduce the presented work to traditional methods, such as utility theory, fuzzy sets, optimization, experimental design, and QFD. Thus, the development of this thesis applies to all these methods as well.

Chapter 5 introduces uncontrollable parameters into the development of Chapter 4. Noise is characterized as a σ-algebra with an uncertainty measure. Given this, Section 5.1.1

develops possible metrics for rating a design configuration, given the noise. A metric is developed as an uncertain integral over the uncertainty space. Section 5.2 applies this general development to the specific noise form of probability (as defined in Chapter 3), and the uncertain integral becomes the traditional expectation of probability theory. Then Section 5.3 applies the general development to the specific noise form of possibility (as defined in Chapter 3), and the possibilistic expectation of possibility theory results. Finally, Section 5.4 applies the general development to the specific noise form of necessity (as defined in Chapter 3), and a new expectation results. When the underlying uncertainty is probabilistic over simple one-dimensional spaces, however, the expectation reduces to traditional class interval methods.

The development through Chapter 5 defines what should be computed by a designer, given a level of formalization. Chapter 6 then accepts this development, and applies it to the preliminary design phase, when a design has been incompletely developed. If only partial specifications have been made on a formal model, Section 6.2 develops a method to propagate the partial specification through a model to the unspecified aspects. This allows a designer to informally interpret the formal results on the unspecified space, to aid specification. Such preference inducement is demonstrated for examples on real parameters, and efficient computational algorithms presented. Also, preference inducement is demonstrated on manifolds with maps, differentiation, and o.d.e.'s. Section 6.3 presents geometric conditions under which the induced preference is invariant over the different combination metrics (as developed in Chapter 4) which can be used. Section 6.4 presents when the induced preference calculation can be used to trivially determine the most preferred design configuration, as calculated in Chapter 4, thus tying the developments of Chapter 4 and Chapter 6. Section 6.5 then develops a method to calculate expected induced preference when there is non-trivial noise, using the developments of Chapter 5.

Appendix A presents historical methods for eliciting initial preferences as used in Chapters 4, 5, and 6.

Appendix B presents proofs of the claims made in Section 6.3.

Chapter 2

Engineering Design Theory

This chapter will discuss a philosophy of engineering design required of the research work. This will provide the justifications for the development presented. No presentation in this chapter can or will be proven in a formal sense; rather, concepts are presented and supported informally. Every formal theory (such as the theory developed in this work) must contain axioms which form its basis; further, the use of any axiom must be justified. This justification occurs in a natural language, by informal arguments. This chapter presents this informal, natural language justification.

The organization of the chapter is by the degree of concept formalization. Those concepts which require no formalization are first presented, and subsequent concepts are introduced as more formalism is developed. Thus, Section 2.1 introduces the "performance improvement principle." Then, Section 2.2 defines the concepts of "informal" and "formal" for the purposes of this thesis. Finally, Section 2.3 characterizes the formal structure assumed.

The work of this thesis describes a single person operating in a design process, and shall be referred to as *the designer*. This work does not describe group efforts, unless specifically discussed.

2.1 Performance Maximization

An assumption made for the subsequent work is that, to the best of their ability, designers will desire to produce the "highest performance" design. This assumption requires more elaboration, in that it is difficult to define what "highest performance" is: "performance" is a term having different connotations. It is assumed that when presented with two alternative solutions to a problem, and if one alternative clearly has the property of "higher performance," then the designer will pursue the alternative with the "higher performance." This is assumed true even when the design problem has no formalization at all. This assumption is stated explicitly as the *performance improvement principle*:

Assumption 2.1 *When given two alternatives, one which is of higher performance, the designer shall pursue the higher performance alternative.*

This is not to say that a design process is always (or only) a search for an optimal solution. This is an overly narrow view of a design process. This assumption simply states that if it is ever reasonable to improve a design, the designer will do so.

Examples of practices which conform to the performance improvement principle will now be presented, to provide justification. For example, the methodology of "value analysis" [54] conforms to the performance improvement principle. Value analysis is a methodology to ensure that the amount of quality per amount of cost is maximized. Performance is therefore amount of quality per amount of cost.

Optimization formulations [123] conform to the performance improvement principle. Optimization formulations determine an overall set of design parameters by minimizing an objective function, subject to additional constraint functions. Performance is defined simply as minimizing the objective function, with the constraints satisfied.

Experimental design techniques also conform to the performance improvement principle. For example, Taguchi's method [26, 124] searches for a design which has minimum sensitivity to noise influences. Here performance is defined as average output across the noise.

Knowledge based design techniques [22, 36, 72] also conform to the performance improvement principle. Such systems search for a design which gives maximal satisfaction, subject to heuristic stopping criteria.

Other examples which may at first seem not to conform to the performance improvement principle in fact do. For example, consider when a designer is restricted by time [180]. A design must be produced within a deadline, and the designer may only be able to develop a design which functions at a minimum level. Given this time restriction, the designer simply desires to satisfy the objectives. The designer also desires, however, to reduce development time. This does not contradict the performance improvement principle. The designer is finding the "highest performance" design considering time as an objective as well. Time becomes a constraint in the design, and has a heuristic target value. Such a design process involves satisfying all the constraints, including time, to the best of the designer's ability.

A difference between development time and other objectives is that time is not required to be modeled explicitly and manipulated. This poses the question of when do objectives need to be explicitly modeled. The discussion so far has remained entirely informal. The next section shall develop the distinguishing aspects of informal and formal models, as required for design activity. The performance improvement principle, however, needs no formalization.

2.2 Engineering Design Models

In a design process, a design is developed to satisfy the needs of a customer. Such needs are typically expressed using informal descriptions [4, 9, 90, 121] which can have many interpretations. For example, many requirements are described in a natural language, where personal connotations enter the communication. The requirement that a design "look sporty" is a natural language requirement with many interpretations. Other aspects may also be informal and not explicitly expressed. For any computation to occur, these informal descriptions must be translated into formal model representations. This section will present the argument that translations between initial, informal descriptions and formal modeling representations occur, and will then make these distinctions explicit.

2.2.1 Informal Descriptions

When a typical design artifact is developed, it is first conceived in informal terms [4, 9, 90, 121]. Requirements that the design must satisfy, for example, may be described in a natural language, complete with personal connotations. Such descriptions are not necessarily linguistic, but could be visual, acoustic, sensual, or aesthetic, for example. Automobiles must have the right "sound" [27]. Rechargeable battery operated designs must recharge "quickly" [11]. Vehicles must have a comfortable "ride" [93]. To enable computation, these informal requirements must be converted into formal model requirements.

Similar to design requirements, a typical designer performs the initial concept modeling in informal terms. "Back of the envelope" concepts are described informally, perhaps with vague drawings, rather than as a formal model on which computation can occur. The first description of a washing machine drive may be "a belt drive." At this stage, computation cannot occur: the description must be made formal. Different people have different perceptions of what is meant by "a belt drive" until it is made formal. Such initial concepts, consisting of desires along with informal descriptions of possible solutions (visual, acoustic, linguistic, or whatever the form), are termed the *informal model* of the design.

Definition 2.1 *An* informal model *is the designer's interpretation of the description of the customer's requirements, and the designer's interpretation of the conceived solutions.*

The actual representation of an informal model, though an interesting research issue, is not part of this work. All that is assumed is an ability to communicate, at some level of competence, descriptions in informal terms, so that they can be translated into a formal model.

Informal models are not precise, are without measure, and have the characteristics of intentions, interpretations, and connotative meanings. Descriptions can be acoustic, visual, or aesthetic, for example, which are (generally) unmeasurable concepts. The term "informal model" is therefore misleading; there is no "model" at all. But the term "informal model" will be used to refer to the interpretations and intentions of the designer.

Each desire in the informal model will be called an *objective*, maintaining the terminology of decision theory [129]. Objectives are unmeasurable (generally). This formalism only discusses the designer's intentions, not any other person's intentions. This assumes that customer requirements are inherited by the designer as objectives. In a design process, the customer's desires become the designer's desires, to the best of the designer's ability to interpret. This is an aspect of the informal model (the informal determination of the designer's desires), not of the formalization process (the conversion of informal objectives into formal model requirements). Short of clairvoyance, designers can only formalize their interpretation of others' desires; they cannot formulate others' actual desires (assuming they exist). Formal techniques for eliciting customer's desires can be used [4, 71]. The designer, however, still adopts a belief in the results of the elicitation; therefore, again the customer's desires are adopted as the designer's desires in a design process.

Consider examples of informal models within current design methodologies. The informal model associated with expert system technology [22, 36, 72] is the initial expert knowledge that the "knowledge acquisition" phase formalizes [151]. Also, the customer's interpretation of the formal results generated by the expert system occurs in informal terms.

Consider grammatical systems which construct feasible design alternatives with design grammars [161, 162]. An informal model can be associated with design grammars, consisting of the intentions reflected by each formal rule in the design language (see Stiny [161] for examples). Also, again the interpretation of the formal results generated by the grammar occurs in informal terms.

The informal model associated with optimization methods [123] consists of the customer's informally expressed requirements which are translated into formal objectives and constraints in the optimization. The informal model associated with matrix methods, such as Pugh's method [126] or QFD [4, 69], are again the customer's informally expressed requirements which are translated into the designer's written interpretation (within the charts).

Given the definition of an informal model of a design, the process and results of formalizing the informal model will now be discussed.

2.2.2 Model Formalization

Informal models must be converted into formal models for computation to occur. This observation on translations of informal models into formal engineering models is fundamental. The work described herein is based on the conjecture that an aspect separating *engineering design activity* from more basic *craftwork activity* is that engineered products require analysis. A craftsman can interpret a need in informal terms, develop a solution entirely with informal terms, construct a solution, and verify that it suffices in the real world. Engineering design activity is not this simple. This assumption will be made explicit:

Assumption 2.2 *Engineering design activity* requires *formal models, which are typically reflections of informal models, on which computation can occur.*

For example, design *optimization* methods require the translation of informal requirements into formal objectives and constraints. Informal solution concepts must be formalized into a domain over which a formal search process will optimize, and the formal constraints will bound. The set of equations is solved as a formal model, and the formal solution is interpreted in informal terms.

Matrix methods require the conversion of informally expressed customer requirements into a formal list of objectives. These are then converted, using the matrix chart, into formal design requirements. Also, the designer's conceived alternatives are formally represented as entries in the matrix chart, and are formally rated.

Consider expert systems for design. In the construction of the expert system, the expertise, expressed in the informal terms of the expert, was translated into the formal language of the expert system. The system then uses its formal rules to deduce new designs, which are then interpreted by the customer in informal terms.

Finally, consider grammatical systems which construct feasible design alternatives with design grammars. In the construction of the design grammars, informal intentions are translated into formal design grammars. The grammars are used to compose new designs, which are then interpreted by the customer in informal terms.

Thus engineering design typically entails the translation of a design from an informal model to a formal model. This idea is depicted in Figure 2-1. A design problem is informally recognized, in the "real world." The problem typically consists of informal descriptions of desires: perhaps natural language requirements. Solution candidates are typically generated as informal descriptions. The design engineer must then convert this informal model into a formal model. Only then can any modern engineering analysis be applied to the problem. With the formal methods, the designer determines a solution (usually iteratively, with refinement of the formal model based on informal desires). Having a solution of the formal model, the designer then translates this formal model into an actual device. This reality is then observed, and interpreted in informal terms by the customer.

This conjecture about engineering design should be tempered by experience. The design of an artifact may require in-depth analysis by a novice designer, whereas a more experienced designer may not require the analysis to derive a solution. It is argued, however, that this

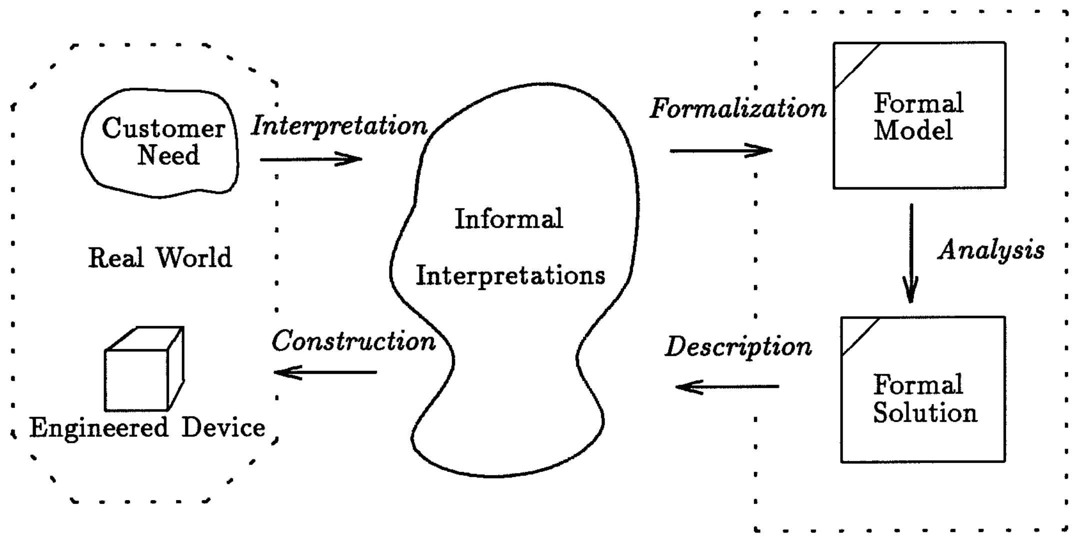


Figure 2-1: Paradigm of data flow through a design process.

represents the ability of the experienced designer to perform intricate design as craftwork. This is not to say it is no longer engineering; it is to say that the activity is beyond what this research work and thesis is intended to provide assistance and foundation.

2.2.3 Discussion

As an aside to the development of these informal arguments as needed for this thesis, one can consider the ramifications of the arguments. Accepting that a canonical engineering design activity involves the transformation of a design from informal descriptions to formal models has consequences. For example, consider the approach of assuming that a candidate model is a problem solution instance from a "design universe" of solutions from the "real world." The "design universe" is a common idea that there is a universe of possible solutions, and if only one had enough time to pursue them all, one could determine "the" absolute best solution to a problem. With the understanding that design solutions and requirements originate from an informal model (rather than from a measured "reality"), it is apparent that such a universe cannot be defined. The "design universe" is an informal characterization and so has no structure (specifically no set structure), and so cannot be defined. This idea mixes informal concepts and formal models. This implies, for example, that no proofs on global optimality of any design problem can ever be made. "Optimality" implies optimal over some set. Here, the set cannot be defined, the "universe" is an informal characterization. The "universe" is anything that the customers feel satisfy their desires. Therefore any discussion about the "universe of all solutions" is not well defined, and must be restricted to natural language discussion. All that can ever be determined is the best performing solution to the formal model.

2.3 Formal Modeling Theory

A distinction between informal concepts and formal models has been discussed; informal characterizations have been informally defined, but a formal model has not been defined. The formal model structure assumed will now be detailed.

2.3.1 Formal Structure

The designer is assumed to have constructed a model of the design problem. The fundamental characteristics assumed to define a formal model are two-fold. The first is that the designer will be able to elucidate the alternatives to choose among: the alternatives are assumed to have the structure of a set [65]. A set is a formal concept. Assuming the formal structure of a set means that when a designer is presented with an object, the designer can determine whether the object is in the collection of possible alternatives. The designer can distinguish between the different alternatives. The second characteristic defining a formal model is that a designer will be able to determine whether any alternative can function as a solution to the problem. When presented with the problem of designing a doorstop, a designer can, with enough analysis, state whether an object can function as a doorstop. This allows the definition of the *set* of possible solutions to be assumed in the subsequent work:

Definition 2.2 *The* design parameter space, DPS, *is the set of considered possible alternative configurations, described using* design parameters, *which the designer has a direct choice over.*

Thus, the DPS is a formal concept, having set structure. This structure is weak (only requiring the two restrictions as discussed above) but nonetheless fundamental. Having identified the DPS means that the DPS circumscribes *all* of the considered alternatives. No other alternative is possible, other than what is formalized within the DPS.

The development of a formal set of alternatives is not sufficient to enable the use of a formal solution procedure to "solve" the design model. Informal interpretations continue to enter a design process in the analysis used to differentiate the elements in the design space.

Evaluations are made on alternatives in the DPS. Such evaluations can be informal. To allow computation, such informal characterizations must be identified, so that a designer knows whether an alternative has the informal property or not. That is, to allow computation, the evaluations must also have set structure.

Thus, to formally evaluate different alternatives in the design space, performance metrics must be developed corresponding to the informal objectives placed on the design problem. Performance metrics are evaluations placed on each alternative:

Definition 2.3 *The* performance parameter space, PPS, *is the dependent set of evaluated performances determined at each point in the DPS, described using* performance parameters.

The word "dependent" in the definition means that the PPS is related to the DPS by a set map. Performance metrics could be simple functions relating the design parameters. For example, equations of stress (performance parameters) relate lengths and material constants (design parameters). The performance parameters could also be results of differential equations, integrations, computer programs, physical experiments, or "IF/THEN" subjective questions. A means of determining the identified performance given an identified design configuration is what is required.

Formal models as above could now be formally "solved." That is, applying the performance improvement principle (Assumption 2.1), one must determine the element in the DPS which has "maximum" performance in the PPS. Many formal models, however, still have more complication. There are also confounding effects which make the map from the DPS to the PPS inexpressible.

Confounding influences can be possible errors in measurement, or manufacturing errors. They could be variations in environment. They could as well be differences in the decisions of agents other than the designer. To incorporate such confounding influences to allow computation, the designer must determine what the influences are. A designer must determine whether the manufacturing or measuring noise is large enough to be of concern. A designer must determine the range of possible environments in which the design must function. These are all examples of an identification requirement: a designer must identify what the confounding influences are. That is, to allow computation, the confounding influences must have set structure.

Definition 2.4 *The* noise parameter space, NPS, *is a set of possible configurations of parameters, described using* noise parameters, *required to evaluate any point in the PPS, which the designer does not have direct choice over.*

Given this characterization of engineering design processes, the complete formal model assumed is now defined:

Definition 2.5 *An* engineering model *consists of a DPS, a PPS, and a (possibly trivial) NPS.*

Inherently in this definition is a set map $f : \text{DPS} \times \text{NPS} \rightarrow \text{PPS}$, implied by the dependency of the PPS. This simply means that given a configuration, the designer must be able to determine the performance.

Discussion

As another aside to the development of a formal model as needed for this thesis, formal models in current design techniques can be considered. A point of this chapter is to make explicit the idea that there is a distinction between informal interpretations and formal models. Formal models have set structure, informal interpretations do not. Some discussions on optimization blur this distinction, which can result in confusion caused by ill-defined problems.

For example, value analysis [54] defines the objective of any design as maximizing value, defined as

$$\text{value} = \frac{\text{quality}}{\text{cost}}.$$

This is a clear mixing of formal models and informal characterizations. "Quality" and "cost" are not formal concepts at the level of discussion intended. What is "quality?" What is "cost?" Cost in time? Cost in dollars (whose dollars)? Perhaps corporate profits? Others suggest societal loss [169]. These are informal concepts: they cannot be formally *optimized. When a specific problem is encountered, formal performance metrics can* be developed to reflect these informal terms. "Quality" and "cost" are then well defined by the formal reflections of these informal concepts. At that point, though, it is not clear that the definition of "value" above is appropriate, that it actually reflects "value." One could define transformations of the formal reflections of "quality" and "cost" to insert in the above "value" definition, and the new formal reflection of "value" is arguably equally valid. Nonetheless, once a value analysis model is formalized, the formulation will consist of a set of alternatives (DPS), and a PPS resulting from the "value" performance metric.

Matrix methods [4, 69, 126] are also formal models. The DPS is formed by a list of alternatives to choose from, and the list of evaluation criteria forms the PPS. Design parameters are the alternatives, and performance parameters are the evaluation criteria. Each alternative, criteria, and ranking reflect informal interpretations, but the actual matrix is a formal device. The list of alternatives and the list of criteria have set structure.

Expert systems which perform design calculations are also formal models. The DPS is the set of alternatives (usually configurations) searched over, and the PPS is the evaluation criteria used [22, 36, 72]. Design parameters are any of the descriptions used to define any solution. Performance parameters are any specific evaluation criteria. Noise is rarely considered in expert systems for design [173]. Many practitioners of expert system technology discuss their work as if the system is operating with informal interpretations. Using the development here, the system rules are considered formal reflections of informal rules used by the experts under the knowledge acquisition study. The position is taken that no informal interpretation occurs in the system at all, and only a formal representation of the deductions of the expert which occurred informally (as viewed by the system builder) are present in the expert system. This viewed is now shared by some AI researchers [151].

Fuzzy sets are another example of formal modeling of informal intentions. The original motivation for fuzzy sets, as developed by Zadeh, was to represent subjective beliefs [200].

This has been developed into "linguistic variables" [76, 201] which are formal words from a considered "term set" which is the set of all words considered, out of the "universe of discourse." Each "term" is transformed into a relation over $[0,1] \subset \mathbb{R}$ so semantic meaning can be computed with fuzzy mathematics, and linguistic relations (as defined by "terms") can be transformed into new resultant "terms." Using the development here, the "terms" are viewed as formal, not as informal. The terms are formal reflections of their informal meaning, as understood by the builder of the fuzzy system. The possible "terms" form a set, and further the "universe of discourse" does not: it is an informal concept. When using any fuzzy linguistic system for design, a designer will informally interpret the formal "terms" allowed by the formal linguistic variable discussion, and will informally interpret any resulting formal "terms" generated.

Other examples of formal models exist in which the performance metrics are not known explicitly, such as experimental design techniques [20, 124]. The set of parameters to choose from, the set of performance metrics, and the set of confounding influences, though, are identified. What is not known is an explicit map from the design and noise parameters to the performance parameters, and so experiments are performed to construct such an explicit map. Experimental design techniques are formal since the parameters identified have set structure.

Finally, grammatical approaches to design are also formal models. The DPS is the set of all possible sentences constructed by the grammars [161, 162]. The PPS is any evaluation criteria used on each generated design (see Stiny [163]) or can be rules used within the language (equality constraints). Noise is rarely considered with design grammars.

2.3.2 Model Completeness

Given that a formal model must be constructed reflecting the design's informal description, the validity of the formal model comes into question. This has been considered previously, and results are now summarized.

There has been development in ensuring the match between the PPS and the informal requirements. Recall customer requirements are typically expressed as informal *objectives*, where each objective expresses a particular desire of the designer. For the PPS to be a valid reflection of the objectives, there must be a one to one correspondence between a performance parameter and an informal objective [129].

For each performance parameter to accurately reflect its corresponding objective(s), it must be comprehensive and measurable [129]. That is, by knowing the value of the performance parameter, the extent that the associated objective is achieved is also known (comprehensive). Thus, at the end of a design process, knowing a performance parameter value will allow the informal interpretation of the achievement level of the associated objective.

The performance parameter must also be measurable. Formal definitions of measurability exist [87], but essentially the performance parameter must be representable on a quantitative scale (a partial order), otherwise the performance parameter is no more formal than the objective itself.

The accuracy of the DPS model as a reflection of the informal descriptions must also be considered. The same restrictions apply as on the PPS. Any informal description which must be designed requires formalization, and each formal representation must be comprehensive and measurable. This ensures the formal model is true to the informal interpretation. For example, if a design is, in informal terms, a "wide belt drive," this can be formalized into a belt with specific dimensions. The formal model is comprehensive, in that a formal result of 2 to 4 inches allows an informal interpretation of the belt width. Also, inches are measurable, "wide" is not.

The noise parameter space is, in the ideal case, not informal. Rather, it is based on measured observations of the environment, which are then directly applied to the formal model. This is rare (some say impossible [150]), however, and usually subjective probabilities of noise must be used, and again the NPS is derived from subjective interpretations. For a perfectly accurate formal model, once again there must be a one to one correspondence between all sources of noise and modeled noise parameters, and each noise parameter must be comprehensive and measurable. There are more complete discussions on the modeling of noise [20] which are beyond the scope of the argument. The point is that all parameters within a formal model must be comprehensive and measurable, and the identification of such parameters is a fundamental feature of any engineering design process.

2.4 Discussion

It may be disheartening to understand that such a large degree of formalization is required to perform computations. In particular, the need for performance metrics at the outset presents problems [183], in that at the outset a designer may not know what the "complete" set of performance metrics (the PPS) should be. In fact, a designer may never know what a "complete" set of performance metrics is. Certainly the designer never knows if the set of considered alternatives (the DPS) is "complete." "Complete" is an informal concept: all possible alternatives which can satisfy customer and designer desires.

It is believed that this chapter circumscribes what a formal design model can provide. It is not claimed that a formal system will be able to perform "true" engineering design. Rather, a formal system can provide insight and assistance to human designers, who operate and discuss in informal terms. The designer has the ability to revise formal models if any new performance metrics are generated (a new PPS is developed), or any new alternatives are conceived (a new DPS is developed). There are always, however, required translations between formal representations and informal interpretations.

Given these definitions of a formal engineering model, this work shall now discuss defining what the designer must determine. Various forms in which the DPS, PPS, and NPS can be cast shall be presented, and the solution criteria to these different forms defined.

Chapter 3

Parameter Forms in Engineering Design

Given the designer has constructed a formal engineering model, it remains to define the objectives of the designer. This chapter shall discuss different approaches to answering this question. Different variables within a formal model have different evaluation criteria associated with them. These different criteria can be categorized to within four informal *parameter forms*: imprecision, probability, possibility, and necessity. Each of these parameter forms will be defined and elaborated in an informal sense. This will establish the basis for the subsequent formal definitions.

3.1 Imprecision

At the start of any formal design process, a designer has developed a formal model. This means the designer has a known list of alternatives to choose from, but the choice has not been made. This form of uncertainty is uncertainty in choice, and shall be called *designer imprecision*.

Definition 3.1 Imprecision *is the uncertainty in choosing among alternatives within a formal model.*

This implies that, at the end of a design process, imprecision does not exist: an alternative has been selected. There remains no uncertainty in choice for the designer.

Imprecision is an informal characterization of a formal model. Having constructed the formal model, one must still include the informal model characterization to determine how

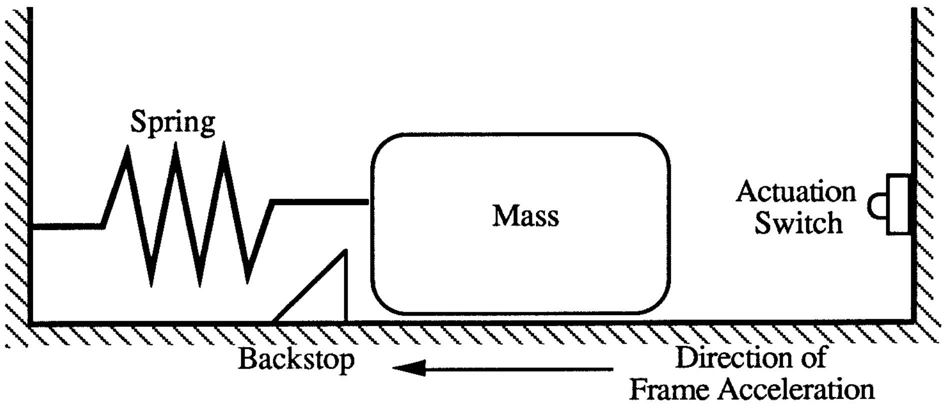


Figure 3-1: Example 1: accelerometer.

the formal model should be manipulated.

For example, consider the design of a uni-directional accelerometer, which indicates accelerations above a threshold acceleration. In this design, when the frame of the accelerometer is accelerated above the threshold, a mass makes contact with an actuation switch, as shown in Figure 3-1. This example shall be denoted "Example 1." It can be modeled as a simple mass spring system, with design parameters of mass, the nominal spring constant, and the overall length. Thus the DPS can be formalized as a subset of $\mathbb{R}^3$. Performance considerations in this example are the time to actuation of the accelerometer (the time it takes the mass to contact the actuation switch), and the required spring pre-load to prevent spurious switch closures. Thus, the PPS can be formalized as a subset of $\mathbb{R}^2$.

Imprecision in this model reflects that, at the start of a design process, the designer does not know what parameter values are desired to be used. Typically, though, the designer may be able to state that certain values of a design parameter will not work, and that some may work better than others. Furthermore, the designer may adjust values of the design parameters to better satisfy the customer, based on the performance evaluations. Thus there is informal interpretation among the parameter values. When the designer has selected values for the parameters (the design model is complete), then there is no imprecision remaining.

3.1.1 Design Trade-Off Strategies

Imprecision is defined as uncertainty in choice. With designs involving multiple parameters, a critical aspect of this uncertainty is the negotiation a designer must consider between the desires for one objective with the desires for another. This can be characterized informally as an *objective resolving strategy*, or "trade-off strategy."

Definition 3.2 *An* objective resolving strategy *is an informal characterization of how a designer intends to trade off different objectives in a design.*

Trade-off strategies are always present in a design process. For example, in the design of a spacecraft solar power cell, a strategy might be to trade-off the performance gains of some objectives (like available power output) to increase the level of other aspects deemed marginal (like stress), to ensure the cell will always function. The term *non-compensating* design strategy, or also *non-cooperating*, will be used to describe a design strategy of trading off to improve the lower performing objectives.

Definition 3.3 *A* non-compensating *strategy for trading off multiple objectives reflects the designer having a desire to improve the weakest performing objective of a design.*

A designer's overall satisfaction will be based on the objective that satisfies least. Other objectives that please the designer more do not compensate for the objective(s) that do not.

On the other hand, a designer may wish to slightly reduce some of the weaker objectives in a design if large gains can be made in the other objectives, that would more than compensate for the slight loss. For example, in the design of a sports car, the designer might reduce the safety margin of some objectives (like stress) to gain in performance of other objectives (like horsepower), even though the stress may already be quite high. The term *compensating* design strategy, or *cooperating*, will be used to describe a design strategy of always cooperatively trading off the objectives to improve the design.

Definition 3.4 *A* compensating *strategy for trading off multiple objectives reflects the designer having a desire to allow higher performing objectives to compensate for lower performing objectives.*

These definitions are informal characterizations; they exist in an informal model. Formal reflections of these definitions will be developed. Also, hybrid forms of these approaches exist and are used, where some portions of a device are designed in a non-compensating manner, and other portions in a compensating manner.

3.1.2 Importance

Another aspect over choice within multiple objective designs is the importance of each objective. A particular objective's value may greatly please a designer, but the objective may also be unimportant to the designer, and so the high performance may not matter.

Therefore, different objectives within a design can carry different importances to the designer. Rating of importance is a relative measure. An objective's importance is relative to the other objectives in a design. Further, the importance can change with changes in the values of the objectives. If an objective attains a poor value, the designer may decide that it is therefore no longer important. Conversely, if an objective attains a high performing value, then the designer may decide to give it high importance.

Assigning importance to objectives is different from selecting an objective resolving strategy. Objective importance can be incorporated into any strategy. Given the desirability and importance of the design objectives, a strategy is formulated to trade off performance of the objectives. Formal methods will be developed that makes this distinction explicit.

3.1.3 Imprecision in Current Methodologies

Imprecision exists in current design practices. For example, imprecision within optimization problem formulations is the uncertainty in choice of the design parameters over which the optimization searches. The design parameter values that do not work are eliminated from consideration by the constraint equations, and all design parameters within the search space are ranked by the objective performance parameter values. When the search process has finished, then a design parameter arrangement has been determined (the arrangement that minimizes the performance parameters): therefore the imprecision no longer exists. Single objective optimization formulations exhibit a non-compensating design strategy: at any given point in the design space, one goal indicates the designer's satisfaction with the design.

Strategies are formally explicit in multi-objective function formulations of optimization problems. Weighted sum techniques [52, 106, 160] are compensating formulations: they minimize higher valued goals simultaneously with lower valued goals. Weighted sums also usually incorporate importance weighting coefficients. Minimax formulations [32, 106] are non-compensating formulations: they minimize the lowest performing (highest valued) goal. Again, importance weighting coefficients are usually incorporated. This clarification shows that any norm minimization scheme [148] applied to multi-objective problem formulations requires two decisions: a goal resolving strategy (norm) must be chosen, and goal weighting coefficients (possibly identity) must be chosen.

The imprecision within expert systems is again the uncertainty in the choice that the system must make to select any configuration. When the system has selected a final configuration for use, the imprecision no longer exists. Strategies and importance is more difficult to explicitly identify in expert systems for design. Indeed, some AI researchers have observed this lack of making explicit goal resolving strategies. Mostow acknowledges

> *"We [AI researchers] need to explicate strategies for how to handle interacting goals." [100]*

In any case, however, the syllogistic reasoning used by expert systems can fixate on particular goals to improve (a non-compensating design strategy), or can reduce one goal to improve another (a compensating design strategy).

Within grammars used for design, the imprecision is the uncertainty in choice over the sentence constructed by the grammars that will be used as a final design. Again, strategies and importance is more difficult to identify, but the design grammars can be interpreted to exhibit both, depending on how the grammars compute the goals. If a goal is fixated upon, a non-compensating strategy is exhibited by the grammar. If a different grammar reduces the performance of one goal to improve another, a compensating strategy is exhibited.

The imprecision within matrix methods is the uncertainty in choice over the candidate alternative that is to be pursued. When the choice has been made, the imprecision no longer exists (in terms of model choice). Design strategies are exhibited by the ranking procedure of the alternatives: for example, by a weighted sum (a compensating strategy), or perhaps by the worst case objective (a non-compensating strategy). Again, weighting coefficients reflecting importance can be used.

3.2 Probability

Unlike imprecision, other parameter forms remain uncertain throughout a design process. For example, measuring limitations always enter a formal model. This represents uncertainty in the ability to determine what is true in a model relative to what it is intended to represent. Such uncertainty is historically well developed [79, 150, 168]. This uncertainty form shall be called *probabilistic uncertainty*, or *random*.

Definition 3.5 Probabilistic uncertainty *is the uncertainty in the accuracy of a model relative to what it is intended to model.*

Thus probability is, again, as with all parameter forms, an informal characterization of a formal model.

A feature of probabilistic uncertainty is that, once identified, the variables that behave probabilistically are not under the control of the designer. Unlike imprecision, a designer cannot choose values for probabilistic variables.

A designer can observe the range of a probabilistic variable, decide it is excessive, and change the process by which the probabilistic uncertainty arose so that the variation is more controlled. For example, a manufacturing process may be altered to allow for tighter fabrication tolerances. This, however, is an example of changing the formal model, not an ability to control a probabilistic parameter within a formal model. Once a particular process is determined, the value of a probabilistic parameter cannot be chosen. The designer cannot choose the exact amount of variation a particular product produced by a particular manufacturing process will be. There will instead be an uncertainty in the variation. This implies that, at the end of a design process, probabilistic uncertainty can remain: manufacturing errors and measuring limitations, for example.

As a specific example, consider the accelerometer example introduced earlier, as shown in Figure 3-1. The spring of the accelerometer may actually be a plate produced by a stamping procedure, in which case the nominal value of the spring constant cannot be accurately produced, due to the manufacturing variability. This error is random, and can be modeled with a noise parameter. Thus the NPS can be modeled as a subset of $\mathbb{R}$, assuming no other relevant sources of noise are in the model.

Another type of modeling error, different from the noise that occurs within the formal model, is the uncertainty in the use of any formal model. For example, in the accelerometer example, the performance parameters may be related to the design parameters by mathematical equations, rather than, for example, experiments on physical models. There is uncertainty in the accuracy of the equations used for the time to actuation and preload, relative to the actual values that will actually be experienced in any particular application of an accelerometer. It is not so clear what formal model of uncertainty should be used to represent this. Traditionally, formal probability has been used [156], but more recent models challenge this and apply belief functions [159] or fuzzy sets [88] for this uncertainty.

Many current formal design practices use probability. Probabilistic optimization [156] introduces noise parameters into optimization formulations, and maximizes the expected performance across the noise. Expert systems for design have begun to address problems with uncertainty [173]. Matrix methods are based on selection in the presence of probabilistic noise [129].

3.3 Possibility

Unlike probability, there can exist other parameter forms that remain uncertain throughout a design process, but are not characterized as random [50, 202]. For example, the limits on choices of a manufacturing engineer (during the fabrication process) over parameters within a design model are uncertain to the designer, but these choices are not random. The choices are not under the direct control of the designer: the manufacturing engineer makes them. The designer can only control their existence in the model (by changing the design so that the model no longer has these variables in it). This parameter form that represents uncertainty in the limits of capacity of a design shall be called *possibilistic uncertainty.*

Definition 3.6 Possibility *is the uncertainty in the limits of capacity within a formal model.*

Thus possibility is, again, as with all parameter forms, an informal characterization of a formal model.

As a specific example, consider again the accelerometer example introduced earlier, as shown in Figure 3-1. There is an additional variable in the design not yet mentioned, and that is the backstop position. Suppose the backstop of the mass can be adjusted to compensate for variations in the spring constant. During manufacture, the spring constant of every accelerometer is measured, and the backstop of each accelerometer is positioned accordingly to meet the specified actuation times. This backstop positioning distance is a possibilistic uncertainty in the design. The designer does not make the choice over the value of the backstop position, it is determined independent of the designer by the manufacturing engineer in response to the spring manufacturing error. This uncertainty is not random. With this backstop positioning uncertainty, the NPS can be modeled as a subset of $\mathbb{R}^2$. This example will be returned to in subsequent chapters.

3.3.1 Tuning Parameters

The possibilistic uncertainty of the accelerometer example is a specific form of possibility that occurs in designs with noise. Specifically, it is a variable that is introduced to overcome the confounding influences of other noise parameters.

Definition 3.7 *Variables representing decisions made in response to noise are termed* tuning parameters, *and their values are determined to increase performance. The values of the tuning decisions are determined* after *the noise on which the tuning decision is based has occurred.*

Tuning parameters are also observed in other engineering problems such as: automotive carburetor idle setting, radio or television signal tuning circuit adjustments, etc. They are characterized by the tuning parameter's ability to compensate for noise.

Tuning parameters are characterized by being set *after* the confounding effects of the noise have occurred. This distinguishes tuning parameters from design parameters, which are set *before* the noise has occurred. Hence design engineers do not set the values of tuning parameters, the manufacturing engineer (or even the customer) sets their values. But when the design parameter values are chosen, the design engineer should consider that a tuning parameter can later be adjusted.

In the terminology of this thesis, the tuning parameters, if any, are a subset of the NPS. This is in agreement with the NPS representing anything in the formal model the designer does not directly choose. Clearly, however, tuning parameters are not representative of probabilistic noise. They are, rather, representative of possibility.

It is not true that a possibilistic uncertainty is always modeled by a tuning parameter. If there had been possibilistic uncertainty associated with the design parameters to limit the designer's choice, then this limitation is not in response to confounding influences. Also notice that, at the end of a design process, possibilistic uncertainty can remain.

3.4 Necessity

Unlike the previous parameter forms discussed, there may exist uncertainty whose range the designer decides must be ensured [182, 185]. For example, performances may have to be ensured above an aspiration level over any manufacturing tolerances, since they could deviate performance from target levels. The parameter form that represents the extent that design aspects must be ensured shall be called *necessity*.

Definition 3.8 Necessity *is any uncertainty within a formal model whose effects must be ensured against.*

Thus necessity is, again, as with all parameter forms, an informal characterization of a formal model.

Necessity is different from the previous forms of uncertainty, in that necessity is not an uncertainty. It is a label associated with the previous forms of uncertainty, indicating that the uncertainty must be ensured against. Thus a parameter can be probabilistic in nature, but have a necessary label. For example, a manufacturing error might be probabilistic, but its effects might be desired to be contained within a specified number of standard deviations.

Necessary parameters occur within current design practices. For example, Motorola has undergone a corporate drive called "6 *sigma*" [1, 68, 91] to contain its manufacturing errors (probabilistic) to within six standard deviations. This is still informal. As a formal example, modern control theory has developed techniques to control parameters such that the worst case noise signal can be controlled [40].

As a specific example, consider again the accelerometer example introduced earlier, as shown in Figure 3-1. Suppose the manufacturing errors of the spring are desired to be contained within a specified number of standard deviations. All produced accelerometers must have the time to actuation and the pre-load performance within a worst case specification. The designer wishes to ensure the worst case. This is a necessary requirement on the design. Alternatively, perhaps the designer instead wishes to improve the nominal performance, or the typical case that will be experienced. This is a non-necessary requirement on the design.

Thus a decision is made regarding necessity and noise: design to ensure the worst case (a necessary model), or design to improve the nominal (a non-necessary model). One cannot do both. More formal examples shall be presented in the subsequent chapters which present formal models of necessity.

3.5 Conclusion

This chapter has presented informal characterizations of different parameter forms. Each parameter form has a different associated optimality criteria. With imprecision, the intent is to maximize the designer's choice. With probability, the intent is to minimize the confounding effects. With possibility, the intent is to maximize the available choice. With necessity, the intent is to maximize the insurance. These are all informal concepts, and will be reflected by formal developments.

These uncertainties can also be characterized informally by considering their effects through a design process. At the start of a design process, imprecision is usually the dominant uncertainty form. The target values of the design parameters are not known. There may be probabilistic and possibilistic noise throughout the designer's model, though these may be reduced by clever choices of the designer. This idea is depicted informally in Figure 3-2. Imprecision is reduced as a design process progresses, until the end of the design process, when a design exists (as a formal model). This by definition means the imprecision is reduced to zero. Probability and possibility can still exist, however, at the end of a design process: manufacturing errors can still exist.

Necessity, again, is different from the other three parameter forms in that, formally, it is simply a label attached to the other forms of uncertainty. This reflects, informally, that what is desired is a different interpretation of the same uncertainty forms. The uncertainty is still characterized as it was without the necessary label. As necessity, though, it must

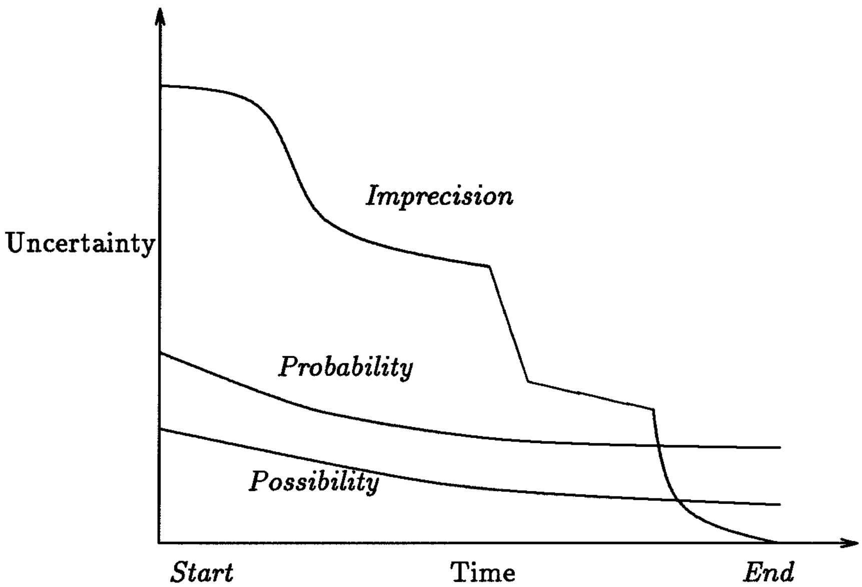


Figure 3-2: Design uncertainties as a design process progresses.

be ensured against. Specific examples will be introduced in the subsequent chapters which will make the distinction clear.

A different informal characterization of the parameters within a formal model are to distinguish between those parameters the designer has choice over (the design and performance parameters), and those that the designer does not (the noise parameters).

The term "noise" is therefore partially used as defined by Taguchi [169], meaning that there are three types of noise observed: external, internal, and variational. External noise errors are due to environmental fluctuations, such as operating temperatures, humidities, etc. Internal noise errors are inherent in the design, such as wear, storage deterioration of materials, etc. Variational noise errors are due to variations in the supplied materials and manufacturing processes.

The term "noise" is expanded beyond this meaning here, however. It also includes effects which are due to choices not made by the designer. Adjustment variables within a design are not under the direct control of the designer, and are thus modeled as being within the NPS. Specific mathematics will be developed, however, reflecting that their character is different from randomness.

Now formal mathematics representing these informal definitions will be developed. The next chapter will deal exclusively with imprecision in design. The subsequent chapter will deal with noise, and the different forms of noise.

Chapter 4

Design Imprecision

This chapter will present a formal methodology for modeling and solving imprecision in design problems, as developed and presented in [117]. Imprecision will be explicitly modeled, and mathematics for manipulating imprecision will be developed axiomatically. Specific examples will then be presented using the developed formalisms.

4.1 Introduction

Chapter 2 introduced and developed the philosophy behind a formal engineering model. This chapter will begin with the assumption that such a formal model has been developed.

Therefore, the designer has developed a formal design parameter space, DPS, consisting of the alternatives to choose among, as defined in Definition 2.2. The DPS will be characterized by *design parameters*, $d_1, \ldots, d_n$. For a design process of selection among alternatives, each d_i represents an alternative, and so the DPS $\simeq \mathbb{Z}_n$ (the finite set of integers up to n). For determination of values in a design model, there are usually multiple parameters each of which could be thought of as a continuum, and so each d_i might thought of as a vector within a DPS $\simeq \mathbb{R}^n$, where the DPS is represented in some basis with coordinates d^i, for example. Recall the DPS always has set structure.

Also, there is a dependent performance parameter space, PPS, consisting of the set of evaluated performances of points in the DPS, as in Definition 2.3. The PPS will be characterized by *performance parameters*, $p_1, \ldots, p_m$. Again, each p_j could be the results of functions, differential equations, integrations, computer programs, physical experiments, or subjective questioning.

4.2 Objective Preferences

Given the formalism so far developed, there is uncertainty within the design model. The designer does not know what design parameter values should be used in the model. This *imprecision* will now be explicitly modeled by constructing a map from the parameters into $[0, 1] \subset \mathbf{R}$ indicating the satisfaction of the designer for values, and will be formally called the *designer preference* for values.

The designer is concerned with the design parameter and performance parameter values. Therefore the designer must consider parameters from the DPS $\times$ PPS. For example, the designer may be satisfied with particular values of a d^i (or p^j), but absolutely reject other values of the d^i (or p^j).

Some of these preference specifications can be made without consideration of any of the other parameters. For example, preferences for a performance metric of material stress can usually be made without considering any other parameter. If the available stress must be less than, for example, about 225 MPa, this can be stated without considering any of the other parameters, such as cost. It is simply known that if the design has stress above this level, the design will not work. If a design must fit within a confined space, and if the design parameters have values larger than the space, then it is known the design will not be acceptable. If the requirements for a design are that it must cost less than a specific amount, this can be stated without considering any of the other parameters. Again, it is simply known that if the design has cost above this level, the design will not be acceptable. On the other hand, other parameters might have their preferences related, and so their preference specifications must occur simultaneously.

Each of these design aspects, whose preference can be stated without considering the other design aspects, consists of groupings of the parameters (both design and performance parameters) and shall be denoted X_k, of which there are N such aspects. A particular X_k shall be termed a *goal*. The designer can specify preferences on each X_k without considering any of the other X_j. These definitions imply

$$\prod_{k=1}^{N} X_k = \text{DPS} \times \text{PPS} \tag{4.1}$$

where $\prod$ denotes Cartesian product. So it is assumed the designer can specify preferences on components X_k of the parameters in the design. Techniques for eliciting preferences have been extensively studied [29, 30, 58, 67, 95, 129, 140, 158, 171, 178, 204], and will not be discussed in the text. Appendix A reviews this work, discussing procedures for eliciting preferences.

It remains to be shown under what conditions the designer will be able to specify preferences. When does a designer have enough information to be able to specify preferences for values? It has been shown [87] that the designer must be able to construct a partial order of preference over each space X_k. That is, when given two alternatives in the space, the designer must be able to state whether one alternative is preferred to the other, or whether the designer is indifferent between the two alternatives.

Given that a preferential order can be constructed over a space X_k, it can be shown [87] that one can also construct a map into the reals that preserves the preferential order. This is true for finite sets, and infinite sets with suitable restrictions on the infinite size.[1] Thus, it is possible to explicitly represent imprecision when the designers can make up their minds about designs. Formally, the map will be defined:

Definition 4.1 *A* preference *is a map* μ_k *from a space* X_k *to* $[0, 1] \subset \mathbb{R}$*,*

$$\mu_k : X_k \to [0, 1]$$

that preserves the designer's preferential order over X_k*.*

A consequence of the definition is that the preferences also specify the valid subset of the DPS and PPS that are to be considered (the support of the μ_ks).

The overall objective is to rate different alternatives in the DPS. This will be achieved

[1] X_k must have an order dense subset that is countable, similar to the rationals dense within the reals. With this restriction, it can be ensured that X_k is small enough for $\mathbb{R}$ to cover.

by combining, in some yet to be determined manner, each of the unrelated preference specifications. The formal problem is to combine the μ_k.

Definition 4.2 *The* overall preference *is a map $\mathcal{P}$ combining the individual preferences into an overall preference,*

$$\mathcal{P} : [0,1]^N \rightarrow [0,1]$$

which reflects the designer's degree of overall satisfaction for a design.

Thus, the preferences specified on the design and performance parameters are combined into an overall preference.

4.3 Objective Weights

Given that preferences for values of goals are explicitly represented, this is still insufficient to fully represent the design problem. As informally discussed in Section 3.1.2, each goal may carry different importance to the designer. In the general case, factors must be included to allow the designer to specify the importance (or *weight*) to be allocated to each goal.

Definition 4.3 *The importance* weightings ω_k *are maps from each* X_k *to* N *different* $[0,1]$ *importance values*

$$\omega_k : X_k \rightarrow [0,1]$$

where each ω_k *reflects the relative importance of goal* k *at a value of* x_k*.*

Recall x_k could represent one or possibly more design or performance parameters together, thus x_k could be a vector $\vec{x}_k$.

The reader is referred to [19, 24, 140, 160] for practical methods on specifying weights; the text will not discuss this aspect of the problem, Appendix A provides a review. It is noted that there are reasons why weighting functions present difficulty [160, 175]. Specifically, any procedure for establishing any importance criteria must be based on approximations from a set of points meaningfully separated, so that the designer can specify the relative importance. To construct this for all points in the DPS, however, may require a limiting process, in which case points in the DPS may not be meaningfully separated. The space is finer than the designer can distinguish. A designer may not be able to distinguish different cases, and so only an approximation can be achieved. Additionally, importance changes with positional changes in the design space, due to the values of all the parameters. Thus, importance is a function of d and p, and can be co-dependent among the values. These concerns are fully discussed in Steuer [160].

To alleviate these concern, the standard solution to the problem of specifying weights is adopted: iteration. That is, it is not assumed the designer can, *a priori*, specify the final goal weights, only preliminary estimates. The designer then gains insight on specifying weights through iteration. In any case, techniques for specifying weights from pairwise comparisons of goals are the "Analytical Hierarchy Process" [140], or the "marginal rate of substitution" [160]. It is noted that, though theoretical issues remain with weighting functions, they are commonly used in practice [4, 5, 69, 128].

Recall assigning weight to goals is a relative measure: a goal's importance is ranked relative to the rest of the goals in a design. This implies the weights should always be normalized by their sum; *i.e.*, the ω_k must be such that

$$\sum_{k=1}^{N} \omega_k = 1. \tag{4.2}$$

One can allow for non-normalized weights; for example, ω_k might be fuzzy. At each point, the non-normal weights must be normalized.

There is another observation on importance: since it is assumed that no goals are trivial or absolutely dominant, the normalized ω_k must be such that, $\forall\ k$:

$$0 < \omega_k(x) < 1. \tag{4.3}$$

The 0 lower boundary condition is actually not strict: the particular goal k then simply drops out of the consideration ($\mu_k^{\omega_k}$ becomes 1). Further, the 1 upper boundary condition is always ensured by the previous normalization requirement.

Recall a final observation is that importance can change with changes in the design. If a goal's preference is low, perhaps a designer may wish to change the goal's importance. This implies weight is a function. Usually it is assumed that slight changes in a goal's value do not induce drastic changes in the goal's importance. This is a continuity requirement on weight; *i.e.*, the normalized ω_k must be such that

$$\lim_{x' \to x} \omega_k(x') = \omega_k(x). \tag{4.4}$$

To defined continuity, this restriction needs a topological structure, which is not assumed in general. If the formalization has such structure, however, Equation 4.4 is appropriate.

These requirements of the weights are summarized in Table 4.1.

4.4 Performance Maximization

Suppose a designer models a design with only imprecision effects (no other parameter forms), and so formulates the design problem as discussed in this chapter. If it is assumed that the design problem and the preferences are are accurately formulated, an objective of the designer might be to solve the design for particular design parameter values.

To determine which design parameters should be solved for, the performance improvement principle (Assumption 2.1) will be invoked. The performance improvement principle states that any better performing alternative will be chosen. Here, performance is explicitly measured as preference. Thus, formally, the problem becomes to maximize the overall preference across the DPS.

Definition 4.4 *The most preferred points* $\{d^*\} \subseteq$ DPS *are defined by*

$$d^* : \mu(d^*) = \sup\{\mu(d) \mid d \in \text{DPS}\}$$

where

$$\mu(d) = \mathcal{P}(\mu_1, \ldots, \mu_N)$$

is the overall preference for a point $d \in$ DPS.

For a determinate design problem, this set will consist of a single element d^*. If $\{d^*\}$ has more than a single element, then the problem has multiple, equally preferred solutions.

This formal definition reflects the designer determining the design that provides the most satisfaction. Of course, the problem is not yet well defined. It remains to formalize the expression of designer "satisfaction" with any given design. $\mathcal{P}$ must be defined. This will now be developed.

4.5 Preference Metrics

This development shall discuss when different functions are appropriate to use as $\mathcal{P}$. First, a set of restrictions that all proposed resolving functions (to use as connectives, or *metrics*[2]) must be consistent (at least those which operate with preferences) is introduced in Table 4.2. Then example functions will be given, and it will be shown when each is appropriate for different problems.

The first restriction in Table 4.2 is a boundary condition requirement. It is a formal statement reflecting that if the designer is completely satisfied with all of the objective values attained by a design (formally, $\forall\ k\ \mu_k = 1$), then the designer is also absolutely satisfied with the design. Similarly, if the designer is completely dissatisfied with the objective values attained, (formally, $\forall\ k\ \mu_k = 0$), then the designer is completely dissatisfied with the design.

The second restriction is a monotonicity requirement. It is a formal statement reflecting that if a designer's satisfaction with an individual objective is raised or lowered, then the designer's overall satisfaction with the design is raised and lowered in the same direction, if it changes at all. Hence, in a multi-component design, if the designer's satisfaction with one component is increased while the satisfaction with the other components remain the same, then the designer's overall satisfaction for the design does not go down. The restriction does not mean the preferences or the performance parameters must be monotonic. If either the preferences specified or the performance parameters used are non-monotonic, then this restriction ensures that $\mathcal{P}$ will monotonically propagate the non-monotonicities.

The third restriction is an annihilation condition. If the designer is completely dissatisfied with the objective values attained by any goal (formally, $\exists\ k : \mu_k = 0$), then the designer will be completely dissatisfied with the design. If the annihilation condition is

[2]Formally, the term metric refers to a *pseudo-metric*. $\mathcal{P}$ does not form a true metric, but rather a pseudo-metric. The uniqueness condition of a true metric is not satisfied: $|\mathcal{P}(x) - \mathcal{P}(y)| = 0 \not\Leftrightarrow x = y$. See Royden [139] for a discussion on metrics.

removed, it would allow a designer to choose a design in which the designer is dissatisfied with an objective. Thus, a design could be chosen which fails an objective. Such a design would, of course, have to do very well in other goals to compensate for this zero rating. Such a result is not useful in engineering design. For example, if a design has failed to satisfy a stress requirement, in that the material stress in a candidate design is excessive (and so the preference for stress is zero), no amount of increase in preference for the cost of the design (by making it cheaper) can compensate. The design fails. Vincent [181] also presents this argument in the case of (non-preference) multi-objective function optimization, as does Harrington [67] in his development of a design's "desirability." This will be discussed at length in Section 4.7 on related work.

The fourth restriction is a continuity requirement. It reflects the idea that as a designer's satisfaction with an individual objective is changed slightly, then the designer's overall satisfaction with the design will change at most slightly. It does not mean the preference for any objective must be continuous. It states only that as any individual objective's preference is continuously changed, the method of combining all the objectives' preferences (reflected by $\mathcal{P}$) will induce only continuous changes in the overall preference, if it changes at all. If some parameters have preference discontinuities, the method of combining them will continuously propagate the discontinuities. Therefore a design will not be abruptly preferred by slight changes in values, unless the parameterizing expressions dictate this.

The last restriction in Table 4.2 is an idempotency restriction. It states that if a designer is equally satisfied with all of the individual objectives in a design, then the designer has this level of satisfaction for the design. This condition is a statement related to rationality.

There are many different definitions of rationality. One definition is that a rational person is one who forms decisions consistent with Bayesian inferencing [176]. Others define rational as one who forms decisions consistent with idempotency, monotonicity, continuity, associativity, and commutativity restrictions [59]. In decision making, rational can also be defined as making decisions consistent with utility theory [58, 129]. It is apparent there are many definitions of rational, since rational is an informal term. An informal definition of irrational behavior is to act in a manner which is against one's objectives [176].

This informal definition must be formalized for the context of design. For designer preference, the informal definition of rationality can be reflected into meaning that an irrational method is to reduce or increase the overall preference beyond what the parameters specify. But even this statement is an informal statement. It will formalized into the restriction of idempotency.

The last restriction of idempotency eliminates any functions that combine preferences in an inherently pessimistic or optimistic manner. Methods that combine individual preferences and artificially reduce or increase the overall preference rank should not be considered: *e.g.*, some of the various triangular norms [43] and power methods [34]. For example, if a design had two goals, each with preference 0.8, one would not expect an overall rating of 0.2, or 1.0, since these results are irrational: they reduced or increased the overall preference beyond what the parameters specified. Idempotency eliminates these possibilities.

The restrictions in Table 4.2 define the set of $\mathcal{P}$ that satisfy the restrictions. The restrictions are minimal, with the exception that the idempotency restriction can be used to derive the boundary conditions.

Definition 4.5 *The set of* idempotent mixed connectives, *denoted* $\mathrm{IMC}([0,1]^N)$, *is the set of all* $\mathcal{P}$ *satisfying the restrictions in Table 4.2.*

Given these restrictions of $\mathcal{P}$ to the set *IMC*, claims can be made of any $\mathcal{P}$ that satisfy them.

Proposition 4.1 *Given* $\mathcal{P} \in IMC([0,1]^N)$, *then*

$$\min\{\mu_1, \ldots, \mu_N\} \leq \mathcal{P}(\mu_1, \ldots, \mu_N) \leq \max\{\mu_1, \ldots, \mu_N\}.$$

Proof. Define $\mu_* = \min\{\mu_1, \ldots, \mu_N\}$. Since $\mathcal{P}$ is monotonic, $\mathcal{P}(\mu_1, \ldots, \mu_N) \geq \mathcal{P}(\mu_*, \ldots, \mu_*)$. Since $\mathcal{P}$ is idempotent, $\mathcal{P}(\mu_*, \ldots \mu_*) = \mu_*$. Thus $\mathcal{P}(\mu_1, \ldots, \mu_N) \geq \mu_* = \min\{\mu_1, \ldots, \mu_N\}$. Define $\mu^* = \max\{\mu_1, \ldots, \mu_N\}$. Since $\mathcal{P}$ is monotonic, $\mathcal{P}(\mu_1, \ldots, \mu_N) \leq \mathcal{P}(\mu^*, \ldots, \mu^*)$. Since $\mathcal{P}$ is idempotent, $\mathcal{P}(\mu^*, \ldots \mu^*) = \mu^*$. Thus $\mathcal{P}(\mu_1, \ldots, \mu_N) \leq \mu^* = \max\{\mu_1, \ldots, \mu_N\}$. ∎

This means any decision made by the designer with $\mathcal{P}$ chosen from *IMC* will result in ratings which are bounded by the worst and best ratings within a design.

Table 4.2 is a list of necessary requirements that any global combination design metric which uses preferences must conform to. Additional constraints will now be placed on the metric, where these additional constraints imply a particular design strategy.

4.5.1 Non-Compensating Design

Equally Weighted Non-Compensating Design

Suppose the designer desires to trade off to improve the lower performing goals (in terms of preference) when selecting a design parameter set d^*; that is, the designer desires to pursue a non-compensating design strategy. Also, assume for the moment that all the individual goals are equal causes of concern to the designer. This implies that, to improve a design, there must be an increase in the preference level of the goal whose preference is lowest.

This means the method to use for combining the multiple preferences is $\mathcal{P} = min$. That is,

$$\mu(d^*) = \sup\left\{\min\{\mu_1, \ldots, \mu_N\} \mid d \in \text{DPS}\right\} \tag{4.5}$$

where $\{d^*\}$ is the most preferred design parameter solution set. Using the *min* as a design metric always rates a design by its worst aspect, meaning the goal with the lowest preference. When searching across the design space, whichever goal has the lowest preference dictates the overall preference. When considering different design configurations (points in the DPS), if the designer can improve the design by switching from one configuration to another, this goal's value will change. Of course, the goal that is the "weakest link" may change with changes in design configuration (changes of position in the DPS). Therefore this metric trades off to improve the lower performing goals.

The additional assumptions made by a designer to restrict $\mathcal{P}$ to *min* should be made explicit. This has been considered previously by others [14, 47], and the complete set of restrictions is shown in Table 4.3, for a two goal design. It has been shown that restrictions 1-3 and 5-8 in Table 4.3 are necessary and sufficient to restrict $\mathcal{P}$ to *min* [47]. Annihilation (restriction 4) can then be derived. It has also been shown with restriction 7 (associativity) replaced by 9 (upper limit) [14]. Thus, in addition to the basic restrictions presented (Table 4.2), if one additionally assumes $\mathcal{P}$ is commutative and obeys the identity restriction, then $\mathcal{P}$ must be *min*. Alternatively, associativity can be replaced by the upper limit

Table 4.1: Goal Weighting Axioms.

$\sum_{k=1}^{N} \omega_k = 1$	(normality)
$0 < \omega_k(x) < 1$ for all k	(boundary conditions)
$\lim_{x' \to x} \omega_k(x') = \omega_k(x)$	(continuity)

Table 4.2: Overall Preference Resolution Axioms.

$\mathcal{P}(0,\ldots,0) = 0 \quad \mathcal{P}(1,\ldots,1) = 1$	(boundary conditions)
$\forall\, k, \mathcal{P}(\mu_1,\ldots,\mu_k,\ldots,\mu_N) \le \mathcal{P}(\mu_1,\ldots,\mu'_k,\ldots,\mu_N)$ iff $\mu_k \le \mu'_k$	(monotonicity)
$\forall\, k, \mathcal{P}(\mu_1,\ldots,\mu_k,\ldots,\mu_N) = \lim_{\mu'_k \to \mu_k} \mathcal{P}(\mu_1,\ldots,\mu'_k,\ldots,\mu_N)$	(continuity)
$\mathcal{P}(\mu_1,\ldots,0,\ldots,\mu_N) = 0$	(annihilation)
$\mathcal{P}(\mu,\ldots,\mu) = \mu$	(idempotency)

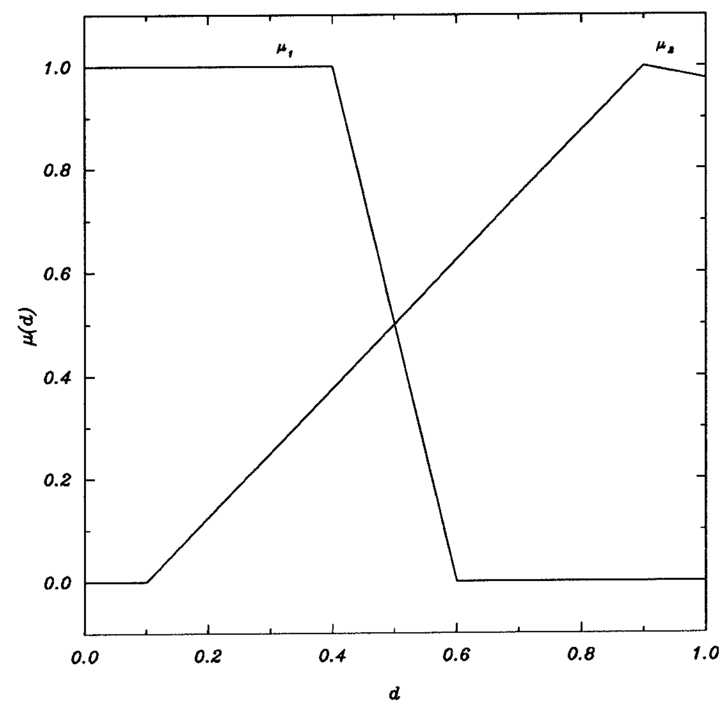


Figure 4-1: Single parameter design with two preference sources.

restriction.

These additional restrictions must be justified using an informal, non-compensating design strategy. Recall the current discussion is restricted to considering the case of equally important goals. In such a case, commutativity is a natural restriction. The order in which a designer combines objectives should not matter. For the same reason, $\mathcal{P}$ should be associative. If a designer combines sub-results which are then combined, the order in which this is done should not matter (for equally weighted objectives).

The non-compensating design strategy implies a design is rated by its worst performing objective. This justifies the final restriction, identity. This restriction is justified by applying the non-compensating strategy definition (Definition 3.3) to the special case of when the other goal (not the lowest performing) has a preference of 1.

The restrictions of Table 4.3 should be accepted by a designer if it is desired to apply the non-compensating design strategy to an equally weighted design problem.

Weighted Non-Compensating Design

Consider now the more general case when each goal does not hold equal importance to the designer. This can be modeled with weights, as discussed in Section 4.3, and applied to the non-compensating design strategy.

For the non-compensating design strategy, the design metric becomes:

$$\mu(d^*) = \sup \left\{ \left(\min \left\{ \mu_k^{\omega_k} \mid k \in \{1, \ldots, N\} \right\}\right)^{\frac{1}{\max\{\omega_k \mid k \in \{1,\ldots,N\}\}}} \,\middle|\, d \in \text{DPS} \right\}. \tag{4.6}$$

This expression reflects trading off the overall performance to gain in the lowest performing goal, with each goal raised to its importance level. In the previous unweighted case (Equation 4.5), each goal had an equal importance of $\frac{1}{N}$. Equation 4.6 reduces to Equation 4.5 when all goals have equal importance ($\forall\ k\ \omega_k = \frac{1}{N}$).

The non-compensating design strategy is affected by the use of importance factors (ω_k) as will be demonstrated graphically for a simple case. Consider a design with just one parameter which has preferences from two sources, as graphed in Figure 4-1. For example, the parameter might be material ultimate strength in a spatial truss. μ_1 might be preference for cost (cheaper materials are more preferred), and μ_2 might be preference for strength (stronger materials are preferred more). As the relative importance of the preferences change (ω_1 goes from an importance of 1.0 to an importance of 0.0 as ω_2 goes from 0.0 to 1.0), the resulting peak preference point changes as graphed in Figure 4-2. The design strategy will choose the value with maximum preference from the resulting combination. For example, with $\omega_1 = 0.75$ and $\omega_2 = 0.25$, the final parameter value chosen will equal 0.44, with a preference of 0.75 (the boxed point in Figure 4-2).

4.5.2 Compensating Design

Equally Weighted Compensating Design

The non-compensating strategy resolution is not always appropriate. If the resulting design is drastically hindered by one parameter and relaxing it a bit greatly increases the others' preference, then the modified design may be considered to produce a higher "overall" performance, even though the lower performing goal was slightly reduced even further. The hindering parameter (the one with lowest preference) should be relaxed and thereby allow other parameters to take on values that substantially increase preference, *i.e.*, a compensating design strategy should be pursued.

This can be accomplished with the use of a product:

$$\mu(d^*) = \sup \left\{ \left(\prod_{k=1}^{N} \mu_k \right)^{\frac{1}{N}} \middle| \, d \in \text{DPS} \right\}. \tag{4.7}$$

This *product*[3] resolution reflects a different design strategy than the *min* resolution presented earlier. Specifically, Equation 4.7 allows higher performing goals to compensate for lower performing goals (in terms of preference). This metric trades off the goals to cooperatively improve the design, and therefore reflects a compensating design strategy (Definition 3.4).

When using $\mathcal{P} = product$, a designer must know any additional assumptions made that restrict $\mathcal{P}$ to the *product*. The restrictions in Table 4.4 are necessary to restrict $\mathcal{P}$ to a

[3] *Product* here refers to the combination as shown in Equation 4.7, *i.e.*, a *normalized* multiplication. It does not mean simple multiplication, without the normalization.

Table 4.3: *Min* Preference Resolution Axioms.

1	$\mathcal{P}(0,0) = 0 \qquad \mathcal{P}(1,1) = 1$	(boundary conditions)
2	$\mathcal{P}(\mu, a) \leq \mathcal{P}(\mu', a)$ iff $\mu \leq \mu'$	(monotonicity)
3	$\mathcal{P}(\mu, a) = \lim_{\mu' \to \mu} \mathcal{P}(\mu', a)$	(continuity)
4	$\mathcal{P}(\mu, 0) = 0$	(annihilation)
5	$\mathcal{P}(a, a) = a$	(idempotency)
6	$\mathcal{P}(a, b) = \mathcal{P}(b, a)$	(commutativity)
7	$\mathcal{P}(\mathcal{P}(a, b), c) = \mathcal{P}(a, \mathcal{P}(b, c))$	(associativity)
8	$\mathcal{P}(a, 1) = a$	(identity)
9	$\mathcal{P}(a, b) \leq \min\{a, b\}$	(upper limit)

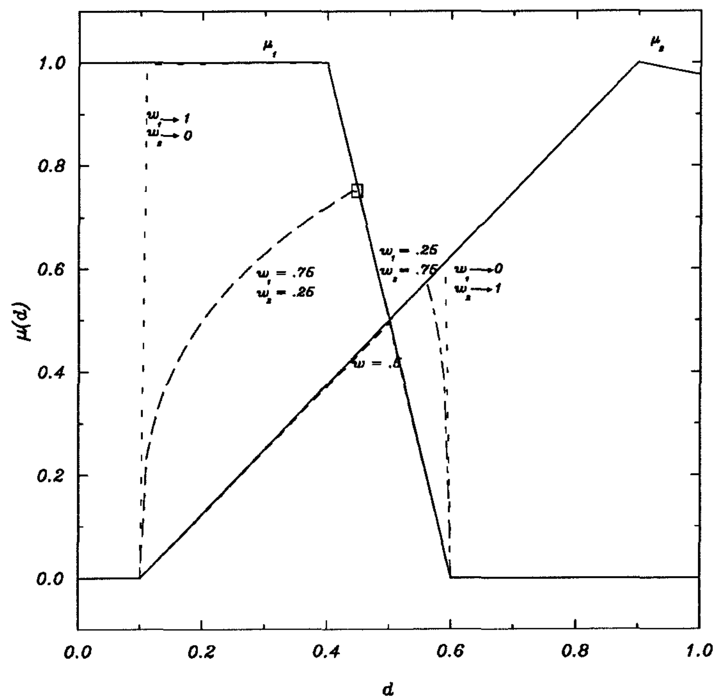


Figure 4-2: Weighted non-compensating design strategy results.

product.

If a designer is to use this compensating strategy, it should be clear why. The additional restrictions of Table 4.4 must be justified, using an informal compensating design strategy. Recall the current discussion is restricted to considering the case of equally important goals. Commutativity is then a natural restriction. The order in which a designer combines objectives should not matter.

For the same reason, $\mathcal{P}$ should be bi-symmetric. It should not necessarily be associative. If a designer combines sub-results which are then combined, the order in which this is done should not matter (for equally weighted objectives). But in a compensating strategy, the result of combining sub-results together carries more weight than an individual goal (when the goals are allowed to compensate in preference). A subset of goals has more capacity to compensate than an individual goal. Thus, associativity is an excessive restriction. Rather, equal numbers of goals within subsets should carry equal weight (as a subset). This is as bi-symmetry restricts (for subsets of order 2).

If a designer has adopted a compensating strategy, then high preference on some goals can compensate for low preference on others. For example, higher preference on the cost of a design (it is cheaper than a different design) can compensate for any decreased preference for performance (the design does not perform as well as a different design). This implies any increase or decrease in preference of any of the goals will change the degree of compensation. Thus, the compensating strategy combination must be strict.

These restrictions should be accepted by a designer if it is desired to apply the compensating design strategy to an equally weighted design problem.

Weighted Compensating Design

For the weighted compensating design strategy, the design metric will use a variation from the previous unweighted case (Equation 4.7):

$$\mu(d^*) = \sup \left\{ \prod_{k=1}^{N} \mu_k^{\omega_k} \;\middle|\; d \in \text{DPS} \right\}. \tag{4.8}$$

This expression reflects trading off the goals cooperatively to gain in the overall performance, with each goal raised to its importance level. In the previous unweighted case (Equation 4.7), each goal had an equal importance of $\frac{1}{N}$. Equation 4.8 reduces to Equation 4.7 when all goals have equal importance ($\forall\ k\ \omega_k = \frac{1}{N}$).

The compensating design strategy is also affected by the use of importance factors (ω_k) as will be demonstrated graphically for the same simple example (Figure 4-1). As the relative importance of the preferences change (ω_1 goes from an importance of 1.0 to an importance of 0.0 as ω_2 goes from 0.0 to 1.0), the resulting peak preference point changes as graphed in Figure 4-3. The compensating design strategy will choose the value with maximum preference from the resulting combination. For example, with $\omega_1 = 0.75$ and $\omega_2 = 0.25$, the final parameter value chosen will equal 0.4, with a preference of 0.78 (the boxed point in Figure 4-3).

This example clarifies that for the same problem with the same preferences and importance factors, the two design strategies selected different solutions. The two strategies traded off the goals in different fashions: compensating or non-compensating. In either case, a goal's importance can be handled within the design strategies.

4.5.3 Other Design Strategies

Hybrid Strategies

Generally, a designer may not wish to trade-off every design component in an exclusively compensating fashion, or in an exclusively non-compensating fashion. A subsystem may need to have its weakest goals maximized, but a different subsystem may need to be cooperatively maximized. For these more general cases, a combination of the two methods (the *min* and the *product*) can be performed, and this is consistent with Table 4.2's restrictions. The sub-design would use the *min* combination of its preference rankings, and this sub-result would use the *product* to be combined with rest of the design.

In the general case, an entire hierarchy of the parameters' preferences would be constructed into the overall metric. Also, this hierarchy could change with positional changes in the design space. A designer may decide to trade off goals in a compensating manner. When one goal becomes sufficiently low in performance, however, the designer may decide to

Table 4.4: *Product* Preference Resolution Axioms.

1	$\mathcal{P}(0,0)=0 \quad \mathcal{P}(1,1)=1$	(boundary conditions)
2	$\mathcal{P}(\mu,a) \le \mathcal{P}(\mu',a)$ iff $\mu \le \mu'$	(monotonicity)
3	$\mathcal{P}(\mu,a) = \lim_{\mu' \to \mu} \mathcal{P}(\mu',a)$	(continuity)
4	$\mathcal{P}(a,0)=0$	(annihilation)
5	$\mathcal{P}(a,a)=a$	(idempotency)
6	$\mathcal{P}(a,b)=\mathcal{P}(b,a)$	(commutativity)
7	$\mathcal{P}(\mathcal{P}(a,b),\mathcal{P}(c,d)) = \mathcal{P}(\mathcal{P}(a,c),\mathcal{P}(b,d))$	(bi-symmetry)
8	$\mathcal{P}(\mu,a) < \mathcal{P}(\mu',a)$ iff $\mu < \mu'$	(strictness)

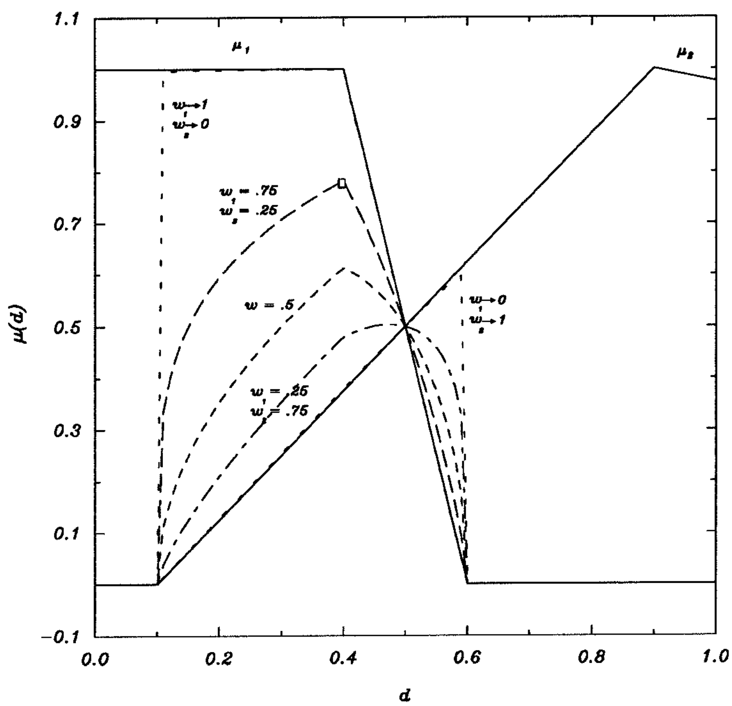


Figure 4-3: Weighted compensating design strategy results.

switch to a strategy of ensuring that goal. Also note that the importance weightings might also change as a design process progresses to reflect the addition of more information.

These aspects make the formulation of a design problem difficult. The solution to this difficulty is iteration. Nowhere is it claimed that a designer can *a priori* specify preferences, weights, and strategies. Through negotiation of the specifications, however, the designer's intentions will become clear. A formal algorithmic procedure for this negotiation has not been presented. Rather, it is felt the designer should make a preliminary specification, and observe the results. Then change the strategy, change the weightings, or change the preferences, and again observe the new results. This iterative process will then provide insight to the designer about the critical objectives, and the unimportant ones (in the designer's interpretations). This will then allow an informed decision. Such an iterative process will be given formal foundation in Chapter 6.

Strategies not Conforming to the Design Restrictions

The past discussion has focused on strategies that conform to the basic imprecision resolving restrictions presented in Table 4.2. There are other methodologies available for combining multiple preferences that do not conform to these restrictions, and will be discussed.

For example, a designer could lexicographically order the goals. In such an order, the goals are prioritized by importance, most important being first priority. Then satisfaction levels α_k are defined for each goal. Preference above the satisfaction level is accepted. The overall preference of a design is rated by the preference of the goal of first priority, unless the preference is above the satisfaction level. In that case, the overall preference is taken as the next prioritized goal, unless, again, its preference is above its satisfaction level. In that case, the overall preference is taken as the next prioritized goal, and so on.

Formally, $\mathcal{P}$ is defined by

$$\mu(d) = \mu_k \ \text{ if } \begin{cases} \mu_i \geq \alpha_i & \forall\, i \in [1, k-1] \\ \mu_j < \alpha_j & \forall\, j \in [k, n]. \end{cases} \tag{4.9}$$

If the design can satisfy all the goals to their satisfaction level, then this lexicographic order is a different reflection of a non-compensating design strategy, as informally described in Section 3.1.1. The preference of the most important goal which is not satisfied becomes the preference for the design, until the final goal is considered. Thus, there is fixation on a particular goal at each point in the design space: the "worst performing" goal is the one with highest importance not satisfying its α_k. In this context, lexicographic orders are non-compensating.

The problem with this formalization is that it is not guaranteed to satisfy the restrictions of Table 4.2. In particular, if the design cannot satisfy all of the objectives, the latter objectives are ignored. Thus an objective may not be satisfied in the final design. The annihilation restriction is not satisfied. Further, changes in d can cause shifts in which goal determines $\mu(d)$. Thus continuity is not guaranteed to be satisfied.

If the lexicographic order is modified by multiplying with a characteristic function to ensure all the preferences are non-zero, then the metric satisfies the annihilation restriction. That is, formally, $\mathcal{P}$ defined by

$$\mu(d) = \left(\prod_{l=1}^{N} \chi(\mu_l)\right) * \mu_k \ \text{ if } \begin{cases} \mu_i \geq \alpha_i & \forall\, i \in [1, k-1] \\ \mu_j < \alpha_j & \forall\, j \in [k, n] \end{cases} \tag{4.10}$$

would satisfy the annihilation restrictions of Table 4.2. $\chi(\mu)$ is a function which returns 1 if $\mu > 0$, and 0 otherwise. Such a formulation would convert $N-1$ goals into constraints. That is, the first 1 to $k-1$ goals restrict the determination to a subset of the DPS where each of these goals have preference above their respective α_k's. The subsequent $k+1$ to N goals restrict the determination to a subset of the DPS where these goals have preference above zero. Such a formulation could therefore result in designs which are just able to work for some goals, but do not attain values near the aspiration levels. It is arguable whether this reflects the designer's desires. In any case, such a metric still fails continuity.

It is also apparent that, at times, designers may wish to use a lexicographic order. Simply saying a designer cannot, because doing so fails to conform to someone else's formal definition of the "proper" method to resolve goals, implies a failure of the definition. But in fact there is a relation between a lexicographic order and the non-compensating design strategy formally modeled by Equations 4.5 and 4.6.

If the designer managed to satisfy the N goals above their respective α_k's, and the $\alpha_k = \alpha$

a constant[4], then the lexicographic formulation result can be achieved more simply by the non-compensating strategy formulation of Equation 4.5. That is, the non-compensating formulation of Equation 4.5 will consider the worst goal preference as the overall preference. Whichever goal has the lowest preference at any point in the design space depends on the location, but as any determination procedure proceeds, in the end the final solution point will have all the goals above α. That is because it is assumed possible to get above the α on the goals, and if one goal were not so, then that goal would be the lowest performing goal, and using Equation 4.5 would improve it. Thus, though Equation 4.10 could be used, Equation 4.5 is more easily formulated.

The construction used to force the lexicographic order to satisfy the annihilation restriction of Table 4.2 was to incorporate a characteristic function applied to each of the individual preferences. This construction could be more generally applied to metrics that fail annihilation. What this construction does is to convert some goals into constraints, by restricting to a subset of the DPS where all preferences are above zero. Such an overall metric can therefore only guarantee that goals are on the boundary of zero preference. This is usually not reflective of a designer's desires.

Instead of a strict characteristic function, a modified characteristic function could be applied, which incorporates the satisfaction level α_k. Defining

$$\mu_k{}' = \begin{cases} \mu_k & \text{if } \mu_k \geq \alpha_k \\ 0 & \text{else} \end{cases} \tag{4.11}$$

then any combination on $\mu_k{}'$ is guaranteed to satisfy annihilation. This presents a new difficulty, however, the specification of the α_k. How does one trade off a specification of α_k versus the resulting overall preference, or versus a different α_j? This trade-off strategy aspect of multi-goal formulations is modeled explicitly with the development of Section 4.4, rather than a designer informally exhibiting a strategy by iterating over the specifications of any α_ks.

The continuity restriction failed with the lexicographic ordering. Perhaps the metric

[4]This is only required to relate the lexicographic order to the non-weighted non-compensating strategy. Each α_k can be different, and the results are as with a weighted non-compensating strategy.

reflected by Equation 4.9 is not an accurate reflection of what is intended by a lexicographic order with imprecision, and instead the metric should "add in" preference from the latter goals in the importance ordering. Therefore, additive metrics should be considered, which, if acceptable, would make lexicographic additive metrics also acceptable.

So, as another example, consider a metric reflected by summing the preferences. This methodology has been extensively studied (decision theory) [58, 87]. The list of necessary and sufficient conditions for an overall preference metric to be additive in the individual preferences are the first five restrictions listed in Table 4.5. The first five restrictions in Table 4.5 are incomplete, they restrict $\mathcal{P} = u$ to being linear. Any linear transformation of a u satisfying Table 4.5 will also satisfy Table 4.5. The boundary conditions are not specified. Taking the boundary conditions of

$$0 \leq u(x^1, \ldots, x^N) \leq 1 \tag{4.12}$$

then u must necessarily and sufficiently be

$$u(x^1, \ldots, x^N) = \sum_{k=1}^{N} \frac{u(x^k)}{N}. \tag{4.13}$$

Thus the first five restrictions within Table 4.5 and Equation 4.12 must be scrutinized.

The essential reason additive metrics are not the best choice for use in design is they fail the annihilation condition of Table 4.2. The additive metric of Equation 4.13 does not satisfy the annihilation condition. If one objective has zero preference ($u(x^k) = 0$), it is not guaranteed that the considered design will be rejected ($u(x^1, \ldots, x^N) = 0$). Others have argued the unsuitablity of additive metrics for design based on this failure, such as Vincent [181] or Biegel and Pecht [18], though not with formulations of preference.

Nonlinear versions of additive metrics are also possible, such as the multiplicative form of utility theory [58, 129]. These formulations obey the first five restrictions of Table 4.5, except for Thompsen's condition, which is replaced by other, less stringent restrictions. Thompsen's condition restricts preferences to be strictly additive [87]. The nonlinear additive metrics, however, still fail the annihilation condition.

In general, the failure of additive metrics (both strict and non-linear) arises from the

Table 4.5: Additive Preference Resolution Axioms.

1	$u(x^k, a) \geq u(x_i^k, a) \Rightarrow$ $u(x^k, x^{\bar{k}}) \geq u(x_i^k, x^{\bar{k}})$	(mutual preference independence)
2	if $u(x_i^1, x_j^2) = u(x_j^1, x_i^2)$ and $u(x_k^1, x_i^2) = u(x_j^1, x_k^2)$ then $u(x_k^1, x_j^2) = u(x_j^1, x_k^2)$	(Thompsen's condition)
3	if $u(\beta, a) \geq u(x_i^k, b) \geq u(\alpha, a)$ then $\exists x^k : u(x^k, a) = u(x_i^k, b)$	(restricted solvability)
4	$\exists n \in \mathbb{Z} : u(na_1, na_2) \geq u(b_1, b_2)$	(Archimedian property)
5	$\exists a, b : u(a, x^k) \geq u(b, x^k)$	(essentiality)
6	$\mathcal{P}(0,0) = 0 \quad \mathcal{P}(1,1) = 1$	(boundary conditions)
7	$\mathcal{P}(\mu, a) \leq \mathcal{P}(\mu', a)$ iff $\mu \leq \mu'$	(monotonicity)
8	$\mathcal{P}(\mu, a) = \lim_{\mu' \to \mu} \mathcal{P}(\mu', a)$	(continuity)
9	$\mathcal{P}(a, a) = a$	(idempotency)
10	$\mathcal{P}(\mu, a) < \mathcal{P}(\mu', a)$ iff $\mu < \mu'$	(strictness)
11	$\mathcal{P}(\mathcal{P}(a,b), \mathcal{P}(c,d)) = \mathcal{P}(\mathcal{P}(a,c), \mathcal{P}(b,d))$	(bi-symmetry)

Archimedian property assumed. The Archimedian property restricts the designer to always allow some objectives to compensate for other objectives [87]. Clearly, this is not always the case. For example, if a design has excessive stress, no amount of decrease in cost can compensate for the excess in stress. The design fails no matter how cheap the cost. Such an example, whose compliance is necessary for design, fails the Archimedian property.

The additive metric just discussed is part of a continuum of possible metrics formed from a range of norms from vector space theory [139]. Perhaps a norm across the goals should be used as an overall metric.

Norms present difficulty for use in general, in that to be a true norm, the set of goals must form a vector space. In particular, if $(F, +, \cdot)$ is a field (F is a set, $+$ is an addition, and $\cdot$ is a distributive multiplication), the set of goals can be considered ordered pairs of F in the usual sense. As a vector space V, though, vector addition $\oplus$ and scalar multiplication $\odot$ must be defined. For each goal equivalent to $\mathbb{R}$, this is clear as $V \simeq \mathbb{R}^N$ with standard vector (component-wise) addition and scalar (component-wise) multiplication.

Each goal is not equivalent to $\mathbb{R}$. Thus, more generally, a norm on V is any function $||\cdot|| : V \to F$ satisfying, $\forall\ v, v_1, v_2 \in V$, $\lambda \in F$:

$$\begin{gathered} ||v|| \geq 0, \quad ||v|| = 0 \iff v = 0 \\ ||\lambda \odot v|| = |\lambda| \cdot ||v|| \\ ||v_1 \oplus v_2|| \leq ||v_1|| + ||v_2|| \end{gathered} \tag{4.14}$$

where $(F, +, \cdot)$ is the field of the vector space $(V, \oplus, \odot)$.

A family of norms can be defined as

$$||v||_p = \left(\sum_{i=1}^{N} (v^i)^p \ \frac{1}{N} \right)^{\frac{1}{p}} . \tag{4.15}$$

This family produces the common norms: $p = 1$ is average of the components, $p = 2$ is RMS average of the components (the common Euclidean length), and $p = \infty$ is the *max* of the components.

In the present context, each component of a vector $v \in V$ is a preference. Thus, $F = [0, 1] \subset \mathbb{R}$, and $V = [0, 1]^N$. With these sets, the field or vector space structure is not apparent. Perhaps using $+ = max$ and $\cdot = min$ would be sufficient. In any case, it is clear that $\mathcal{P}$ cannot be a norm on any vector space constructed over the set of goals $[0,1]^N$. The annihilation condition directly conflicts with positive definite condition of a norm (the first requirement). Thus, $\mathcal{P}$ is not a norm, and using any norm on the preferences would allow a solution that may not satisfy a goal.

Other strategies can also be developed that are not explicitly covered in this text. There are infinitely many different ways to trade off objectives. Therefore, guidelines should be established for any metric. The restrictions in Table 4.2 are seen as minimal for design. Particular combinations can be invoked, but they should conform to Table 4.2.

4.6 Examples

To make these concepts clear, two more examples will be presented. The first (Example 2 of the thesis) is from the conceptual design phase, and involves making a selection between alternative preliminary candidates using the common matrix technique. The incorporation of imprecision is demonstrated, along with the effects on the results. The second example (Example 3 of the thesis), is from the parametric design phase, and involves the determination of shape parameters in an air tank design. This example makes use of the common nonlinear programming machinery. Again, the incorporation of imprecision is demonstrated, and the effects on the results.

4.6.1 Example 2: Conceptual Design

Consider the design task involving a selection between two candidate concepts. The candidates are to be used for assembling items in a manufacturing production line. The first candidate design is a special purpose mechanism, the other is a general purpose robotic arm. Thus, the design problem has been formalized into a DPS = {robot, mechanism} $\simeq \mathbb{Z}_2$, a two alternative design space.

The decision criteria for determining which candidate to pursue into the subsequent design stages are listed in Table 4.6. Thus, the dependent PPS consists of seven criteria, and PPS $\simeq \mathbb{Z}_7$. At this point in the design stage, there is no explicit map between the DPS and the PPS, other than the matrix itself. Designer preference and importance ranks are directly stated. Listed in Table 4.6 is each criterion's importance (on a scale of 0 to 5), and each candidate's ability to satisfy the criterion (on a scale from -5 to 5). Background and details of matrix methods are discussed in [5, 126].

One could use the standard weighted sum matrix analysis [5], which is reflected by an additive metric (similar to Equation 4.13). Using a weighted sum, the mechanism candidate produces an overall rank of 54, and the robot candidate produces an overall rank of 43. This technique guides the designer to pursue the mechanism.

Using the Method of Imprecision, the ranks are normalized by the range of the ranking. As well, the importance ratings are normalized by their sum. The results of this calculation are shown in Table 4.7.

Strategies for resolving these multiple attributes of the candidates can be invoked. Assume that the designer wishes to trade-off the criterion in a compensating fashion, meaning that the designer is willing to measure the overall preference of each alternative based on a composite of its attributes. This implies that some goals with high preference can compensate for others with low preference. Then Equation 4.8 can be used to combine the preferences. Doing so results in a rating of 0.65 for the mechanism, and 0.63 for the robot. Again, the mechanism is determined to be the most promising candidate to pursue.

Now instead, assume that the designer wishes to trade-off the goals in a non-compensating fashion, meaning that the designer will measure the overall performance for each alternative based on the worst (lowest preference) attribute. This implies that the attributes that perform well cannot compensate for those that perform poorly. Then Equation 4.6 can be used to combine the preferences. Doing so results in a rating of 0.40 for the mechanism, and 0.48 for the robotic arm. With this metric, the robot is determined to be the most promising candidate to pursue. This result is different from the cooperative trade-off strategy result. The new choice was caused by the mechanism being rated poorly at development cost, which was not compensated for by the other superior ratings of the mechanism.

Note the standard matrix method resolved the most promising candidate by selecting the one with the highest weighted performance average across the goals. It did not do so by rating each candidate by the worst aspect. Therefore the standard weighted sum matrix

Table 4.6: Example 2: Raw Designer Rankings.

Criteria	Importance	Mechanism	Robot
Ease to satisfy quantity rate	4	5	−1
Ease to ensure operator safety	4	−1	0
Development cost	5	−1	3
Ease to ensure production reliability	5	3	0
Ease to ensure size constraints	2	4	2
Ease to do design by production time	3	0	4
Ease to ensure production quality	4	5	4

Table 4.7: Example 2: Imprecise Designer Rankings.

Criteria	Importance	Mechanism	Robot
Ease to satisfy quantity rate	$\frac{4}{27}$	1.0	0.4
Ease to ensure operator safety	$\frac{4}{27}$	0.4	0.5
Development cost	$\frac{5}{27}$	0.4	0.8
Ease to ensure production reliability	$\frac{5}{27}$	0.8	0.5
Ease to ensure size constraints	$\frac{2}{27}$	0.9	0.7
Ease to do design by production time	$\frac{3}{27}$	0.5	0.9
Ease to ensure production quality	$\frac{4}{27}$	1.0	0.9

technique invokes a compensating goal trade-off strategy, informally similar to Equation 4.7. The developments described here allow for a variety of design strategies. Combinations of non-compensating and compensating strategies could be used for different sub-arrangements of the goals, and then these sub-arrangements combined with either a compensating or non-compensating strategy, depending on the designer's judgments.

4.6.2 Example 3: Parametric Design

The example presented below considers a pressurized air tank design, and is the same problem as presented in Papalambros and Wilde [123], page 217. The reader is referred to the reference [123] to see the restrictions applied to the problem to permit it to be solved using crisp constraints and various optimization techniques (monotonicity analysis, nonlinear programming). The example is simple and was chosen for that reason, and also the ability of its preferences to be represented on a plane for a visual interpretation.

The design problem is to determine length and radius values in an air tank with two different choices of head configuration: flat or hemispherical. See Figure 4-4.

There are four performance parameters in the design. The first is the metal volume m:

$$m = 2\pi K_s r^2 l + 2\pi C_h K_h r^3 + \pi K_s^2 r^2 l. \tag{4.16}$$

This parameter is proportional to the cost, and the preference ranks are set because of this concern. Another performance parameter is the tank capacity v:

$$v = \pi r^2 l + \pi K_v r^3. \tag{4.17}$$

This parameter is an indicator of the design's principle objective: to hold air. This parameter's aspiration level ranks the preference for values. Another parameter is an overall height restriction L_0, which is imprecise:

$$l + 2(K_l + K_h) r \leq L_0. \tag{4.18}$$

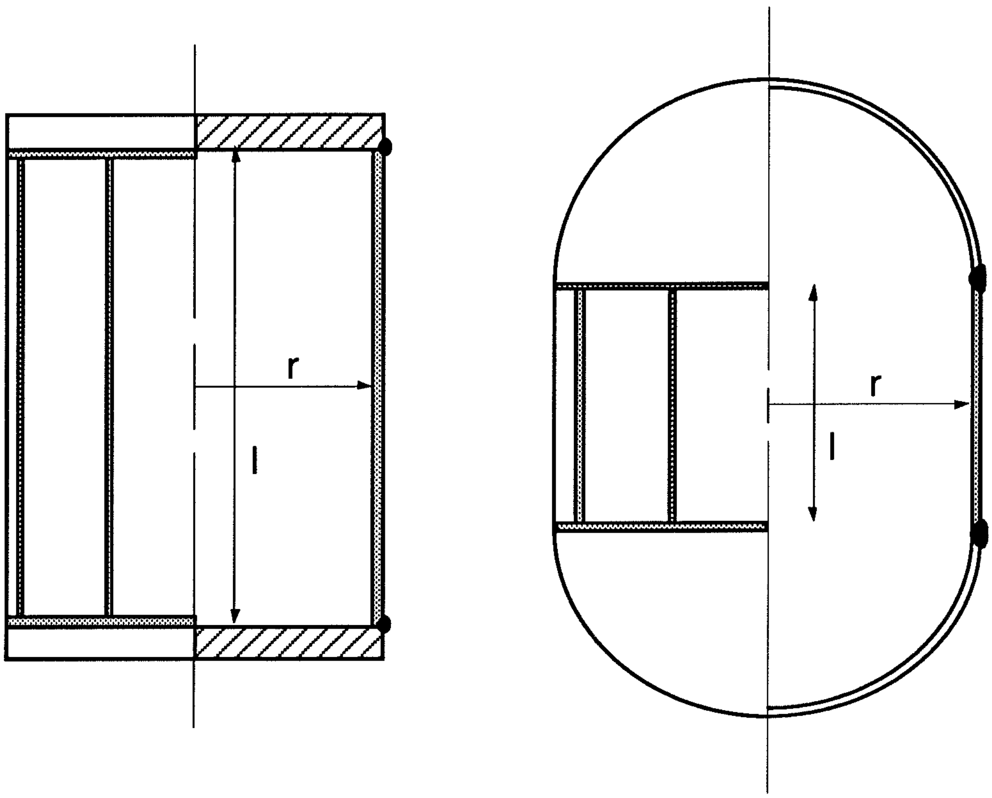


Figure 4-4: Example 3: flat and hemispherical head air tank designs.

Finally, there is an overall radius restriction R_0, which is also imprecise:

$$(K_s + 1)r \leq R_0. \tag{4.19}$$

The last two performance parameters have their preference ranks set by spatial constraints.

The coefficients K are from the ASME code for unfired pressure vessels. S is the maximal allowed stress, P is the atmospheric pressure, E is the joint efficiency, and C_h is the head volume coefficient.

$$K_h = \begin{cases} 2\sqrt{CP/S} & \text{flat} \\ \frac{P}{2S-.2P} & \text{hemi} \end{cases} \tag{4.20}$$

$$K_l = \begin{cases} 0 & \text{flat} \\ 4/3 & \text{hemi} \end{cases} \tag{4.21}$$

$$K_s = \frac{P}{2SE - .6P} \tag{4.22}$$

$$K_v = \begin{cases} 0 & \text{flat} \\ 1 & \text{hemi.} \end{cases} \tag{4.23}$$

This example's design space is formed by the 2 parameters to choose, l and r, and so the DPS $\simeq \mathbb{R}^2$. The performance parameter space is formed by the 4 performance parameters m, v, L_0, and R_0, and so the PPS $\simeq \mathbb{R}^4$. The preferences for values of these parameters are graphed in Figures 4-5 through 4-10 for the hemispherical design; the flat head design space is similar.

The problem is to find the values for l and r that maximize overall preference. For comparison, both a non-compensating and a compensating strategy will be presented and contrasted below. Both consider all goals to be equally important.

For the non-compensating design strategy, l^* and r^* are to be found, where

$$\mu(l^*, r^*) = \sup\left\{\min\{\mu_l, \mu_r, \mu_{v(l,r)}, \mu_{m(l,r)}, \mu_{L_0(l,r)}, \mu_{R_0(l,r)}\} \mid (l,r) \in \mathbb{R}^2\right\}. \tag{4.24}$$

This will find the l^* and r^* by trading off the goals to improve the lowest performing goal (in terms of preference), even though the design parameters and performance parameters

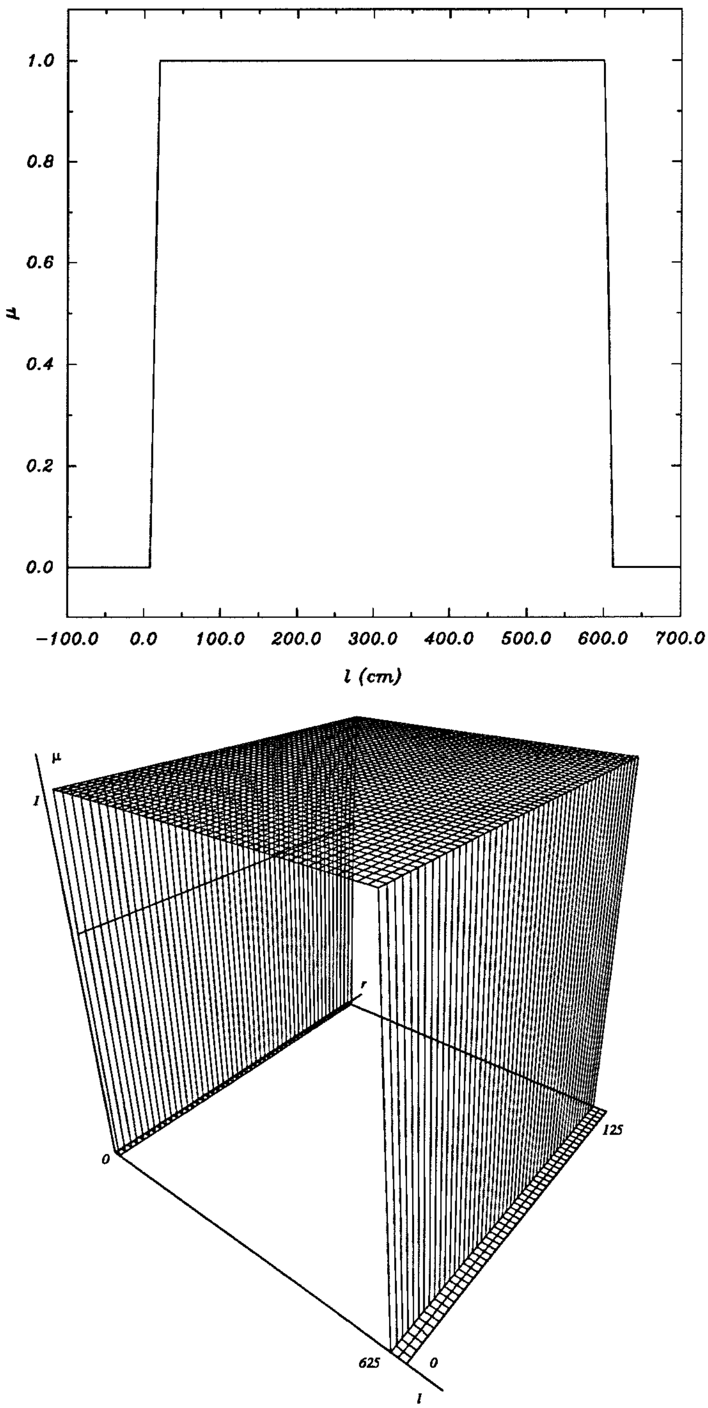


Figure 4-5: Example 3: length l preference.

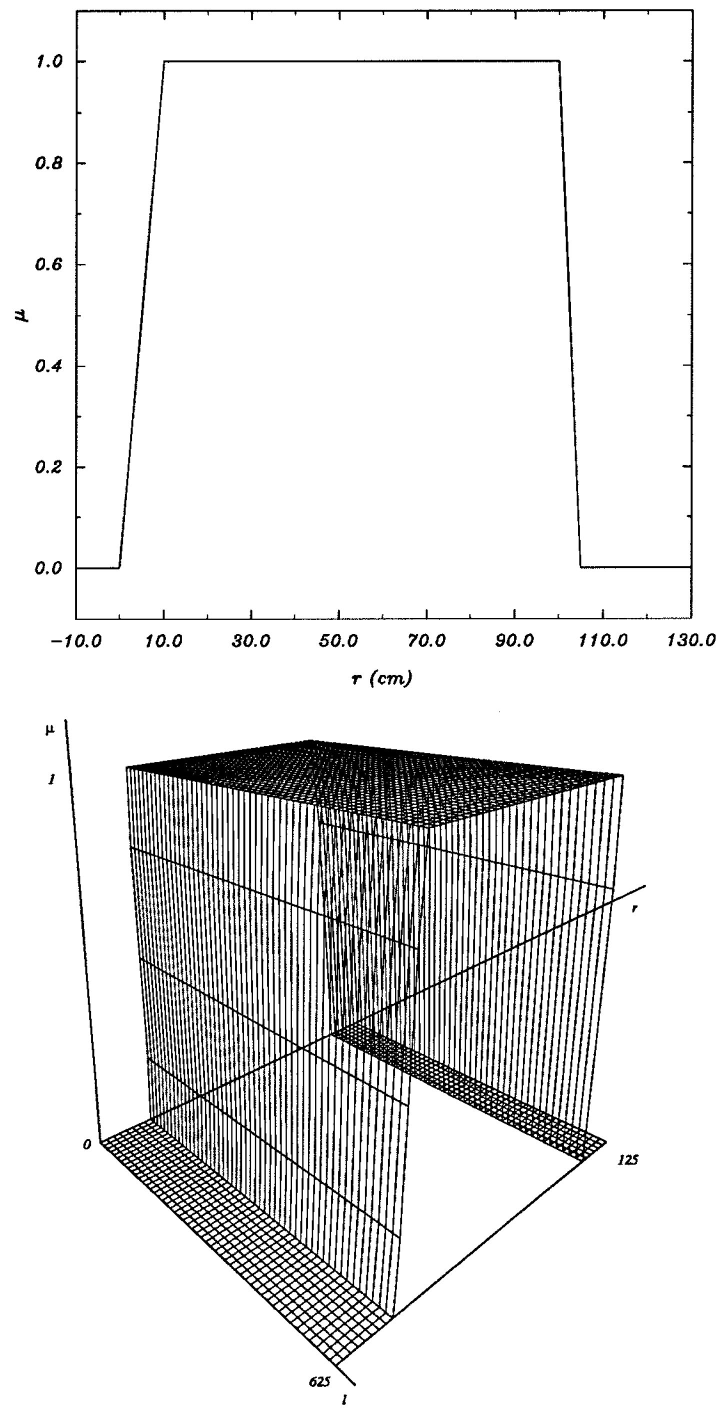


Figure 4-6: Example 3: radius r preference.

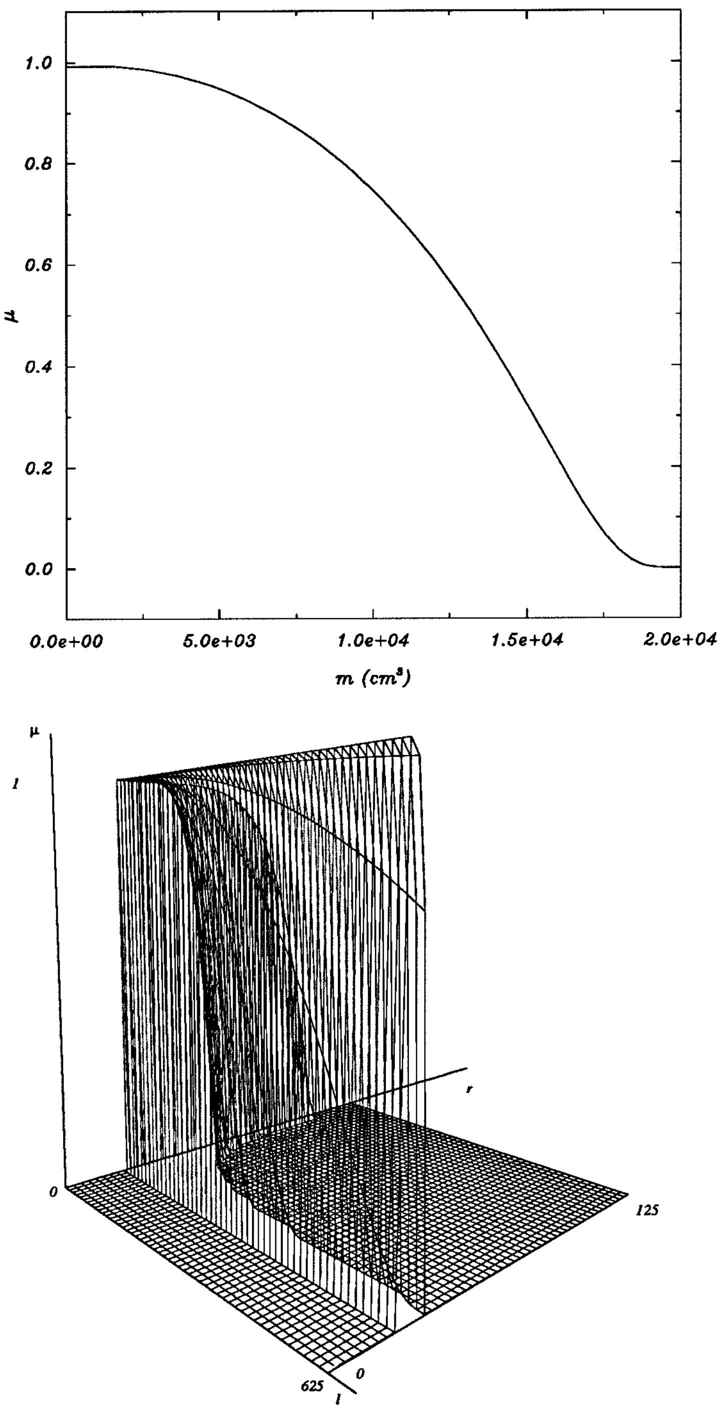
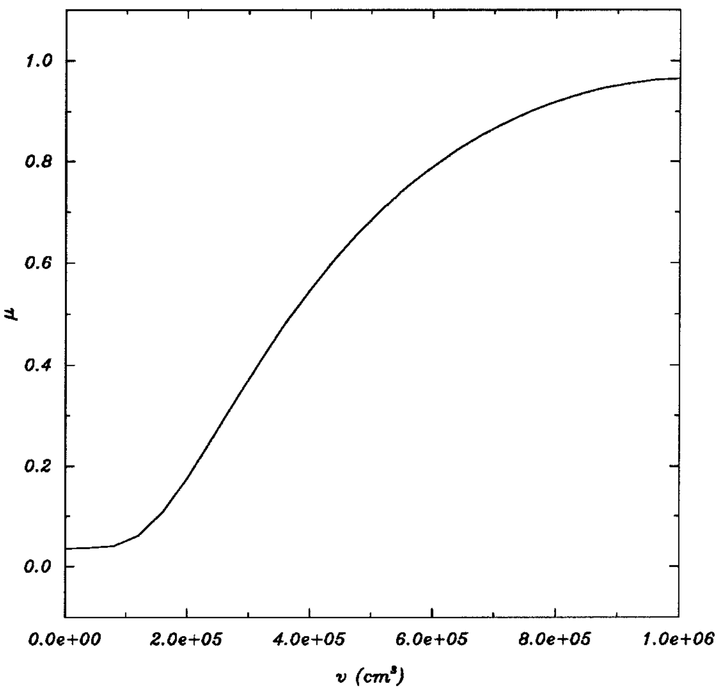


Figure 4-7: Example 3: metal volume m preference.



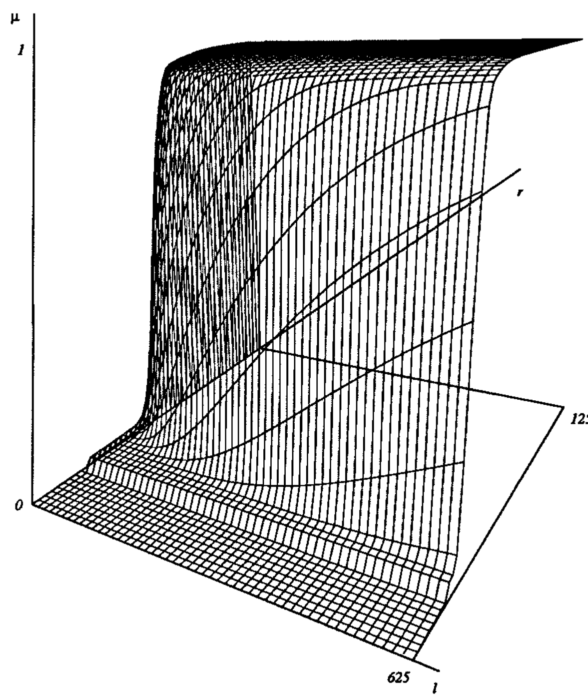


Figure 4-8: Example 3: capacity v preference.

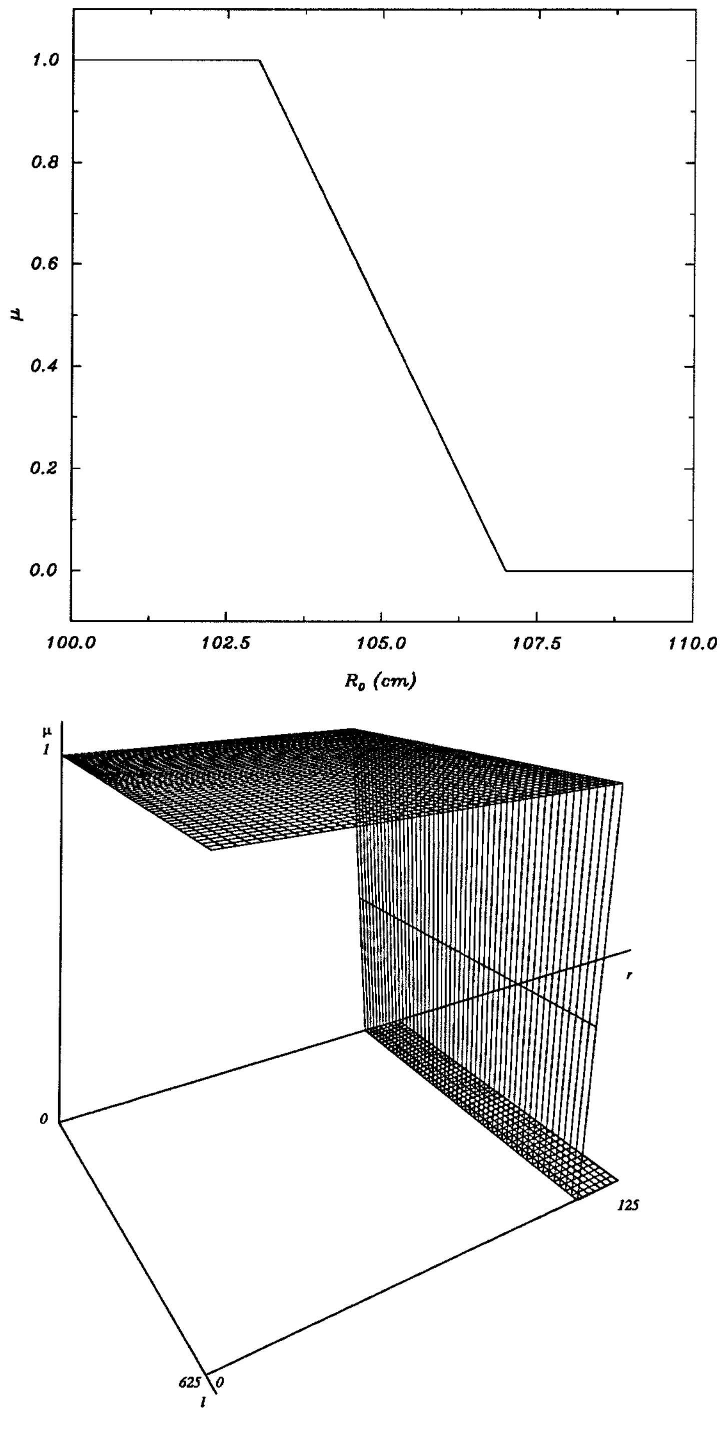


Figure 4-9: Example 3: outer radius R_0 preference.

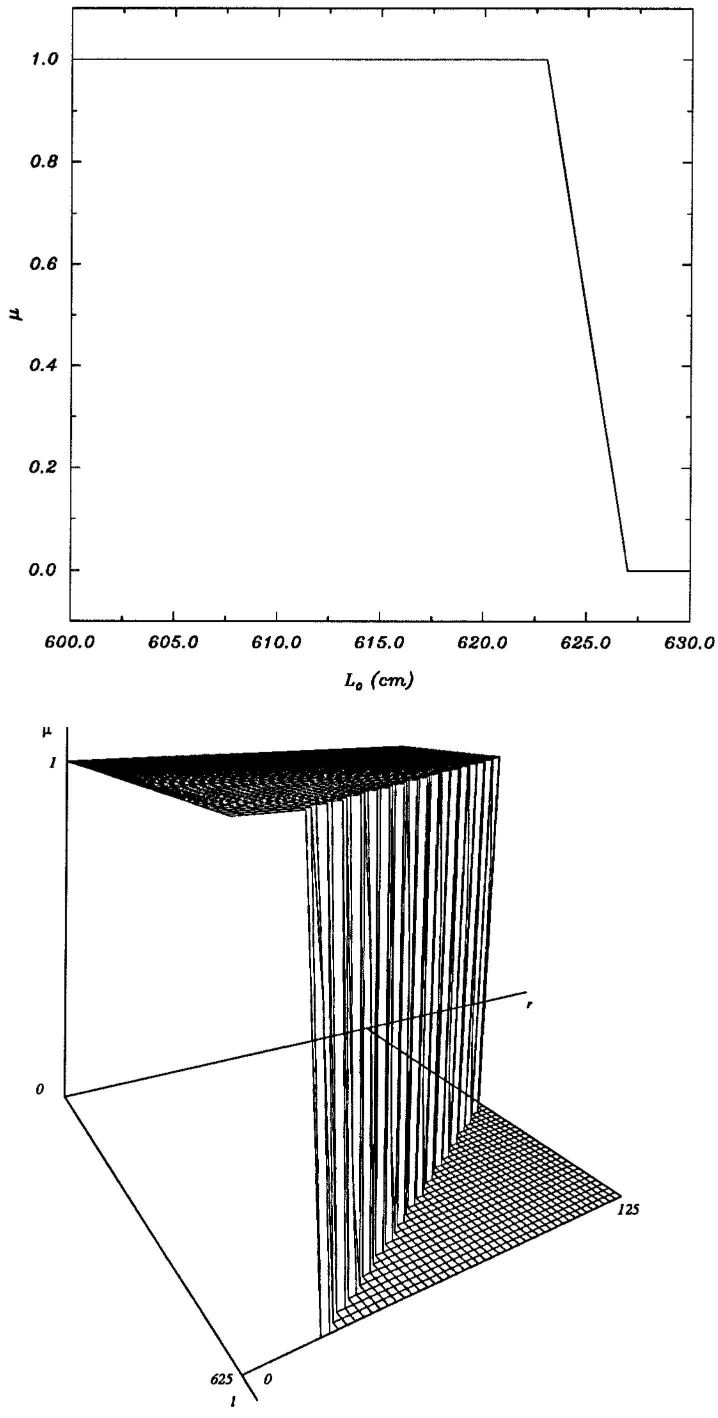


Figure 4-10: Example 3: outer length L_0 preference.

are incommensurate with each other.

For the compensating design strategy, the problem to be solved is to find l^* and r^* where

$$\mu(l^*, r^*) = \sup\left\{ \left(\mu_l \cdot \mu_r \cdot \mu_{v(l,r)} \cdot \mu_{m(l,r)} \cdot \mu_{L_0(l,r)} \cdot \mu_{R_0(l,r)}\right)^{1/6} \,\middle|\, (l, r) \in \mathbb{R}^2 \right\}. \tag{4.25}$$

This will find the l^* and r^* by trading off the goals cooperatively among each other, allowing the higher performing goals to compensate for the lower performing goals (in terms of preference), even though the design parameters and performance parameters are incommensurate with each other.

The preference combination results are graphed in Figures 4-11 through 4-14. For the non-compensating design strategy, the *min* of each individual preference across the design space is the resulting surface. This is graphed in Figures 4-11 and 4-12. The surface's maximum value in μ is the solution point to use (the most preferred l and r). For the compensating design strategy, the individual preference surfaces are multiplied together as a product of powers for all points on the l, r plane. This is graphed in Figures 4-13 and 4-14. These overall preference surfaces should be compared with the individual goals' preferences graphed in Figures 4-5 through 4-10 to observe the relations between individual goals' preferences over the design space, and the end resulting preference surface.

As can been seen, the compensating strategy will produce higher overall preference than a non-compensating strategy, and the two strategies will result in different solution design parameter values for the design: different l^*,r^* have the highest μ on the overall preference surfaces of Figures 4-12 and 4-14 (hemispherical head design), and likewise for Figures 4-11 and 4-13 (flat head design). The non-compensating design strategy sacrificed the cost (m) to ensure the capacity (v). Designing with a compensating strategy did the reverse: reduced the cost (m) at expense of the capacity (v).

This differs from the results of the various problem formulations presented in Papalambros and Wilde [123]. For example, the non-linear programming formulation solves the problem by minimizing the metal volume with the rest of the goals as crisp constraints. The formulation presented above allows the constraints to be elastic, as graphed in Figures 4-5 through 4-10, so the final design parameter values determined are different than if crisp constraints had been used. If the example had selected step functions for preferences

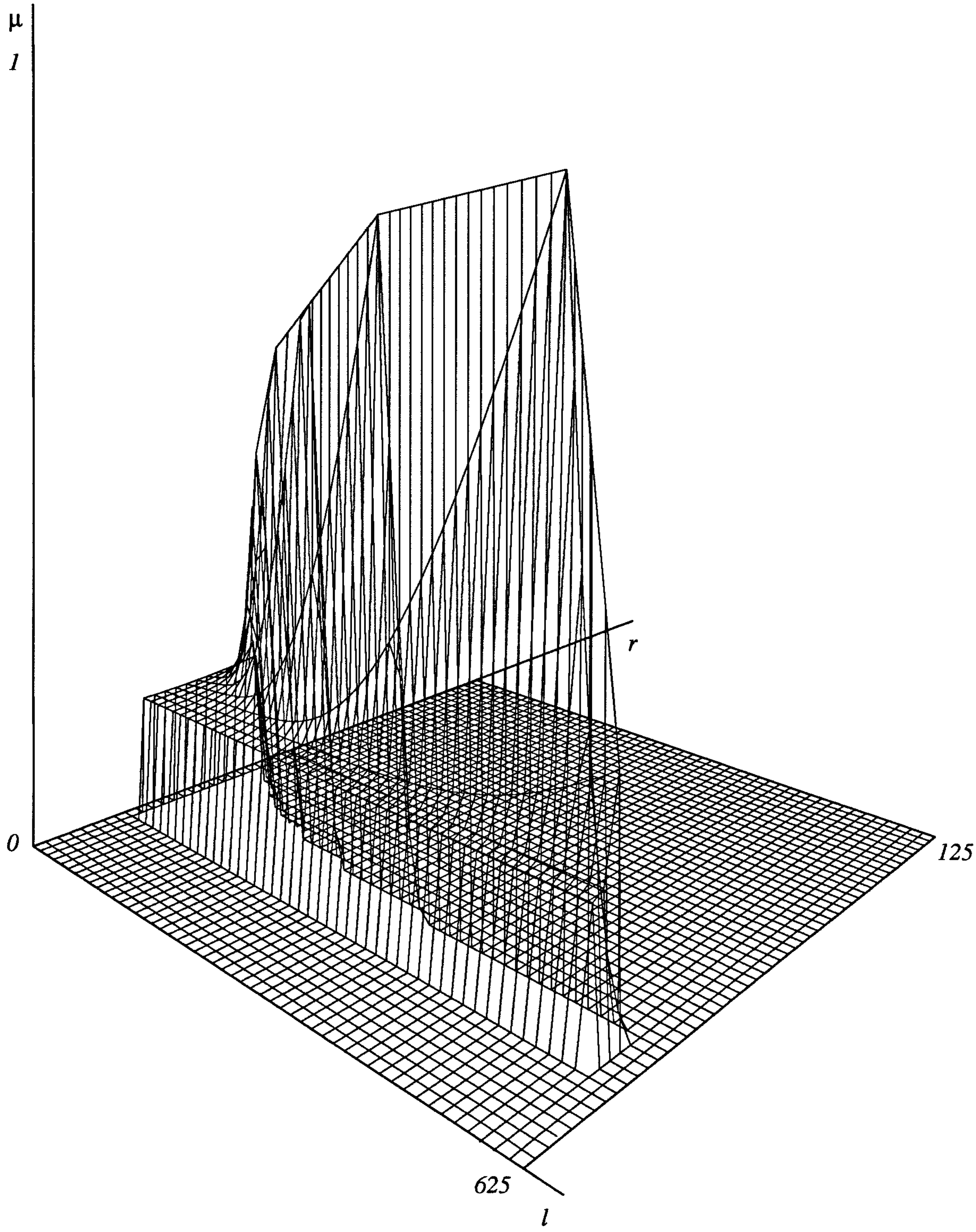


Figure 4-11: Example 3: flat head tank non-compensating design strategy results.

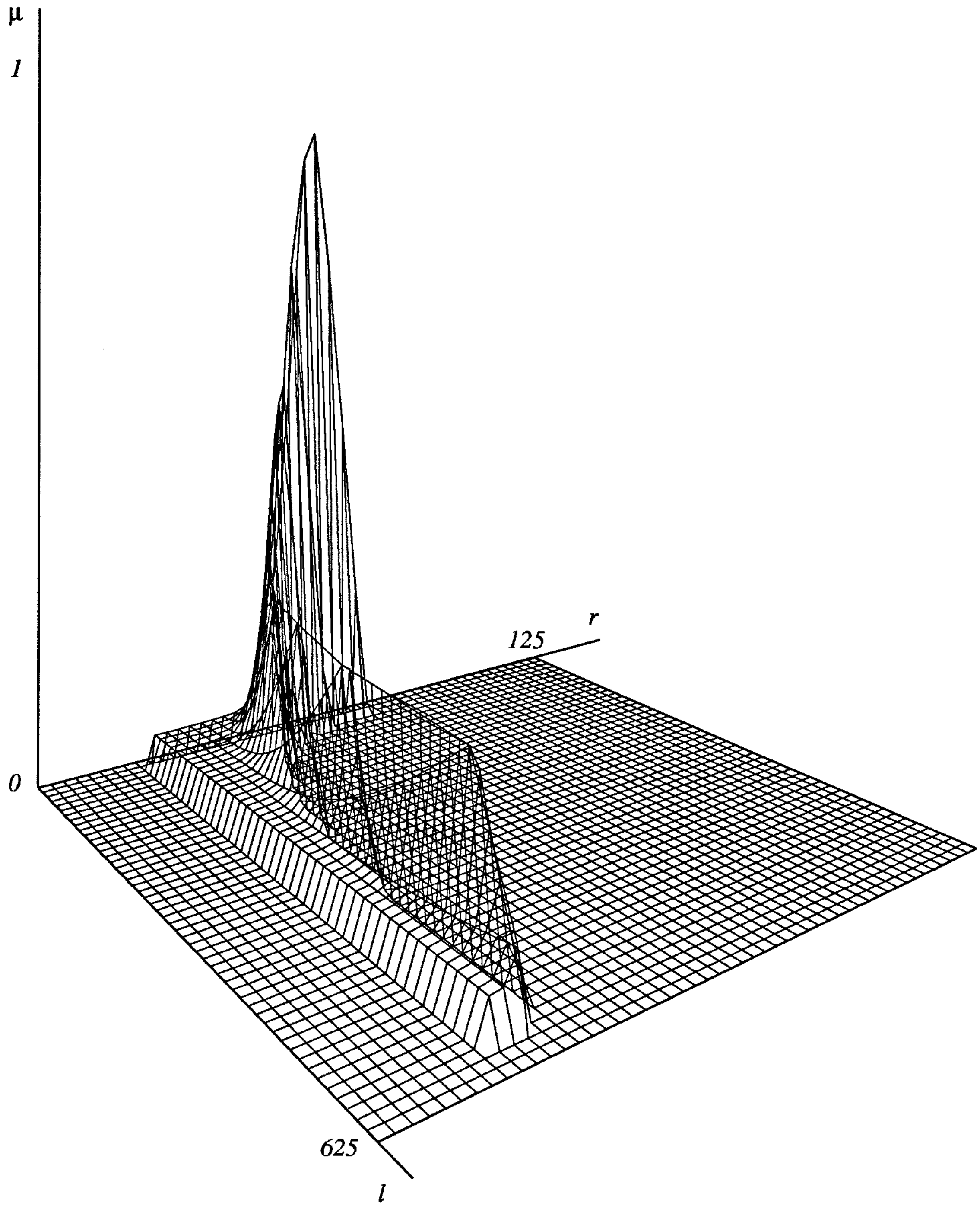


Figure 4-12: Example 3: hemi head tank non-compensating design strategy results.

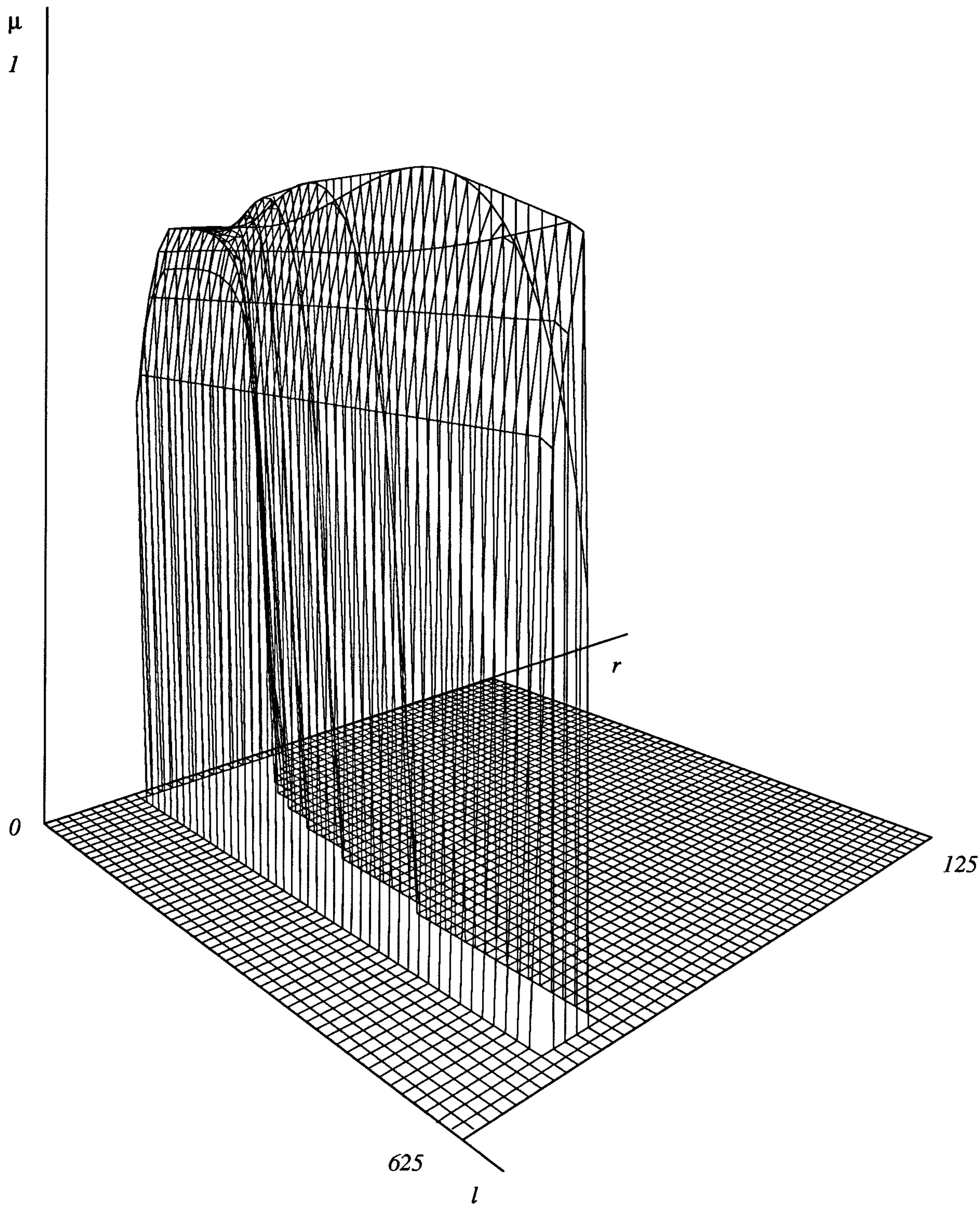


Figure 4-13: Example 3: flat head tank compensating design strategy results.

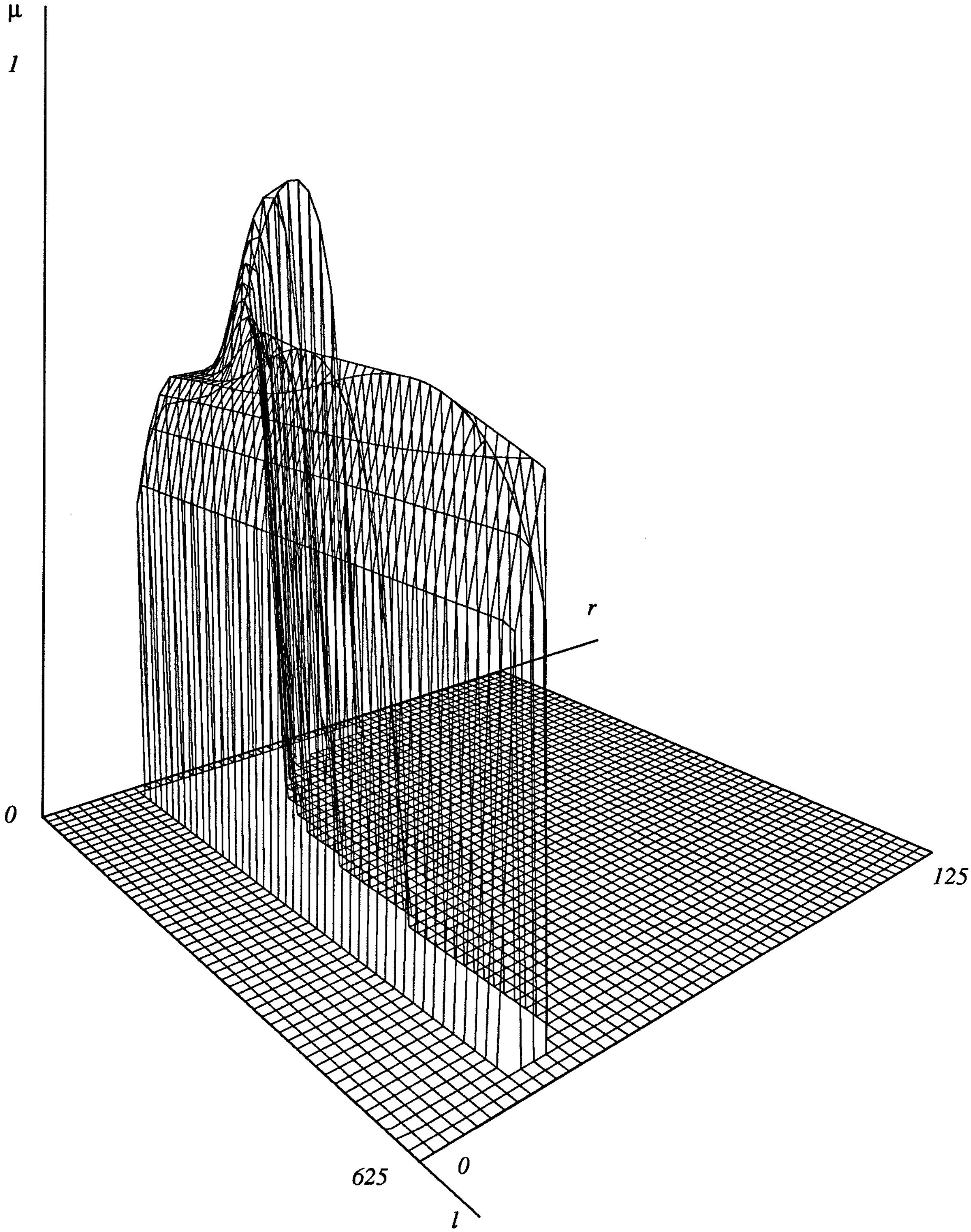


Figure 4-14: Example 3: hemi head tank compensating design strategy results.

on the constraint performance parameters, the imprecision results would reduce to the non-linear programming solution for any strategy. This is because, with step functions for the constraint parameters, only one parameter (m) dictates the preference, and so the issue of trade-off between goals is not applicable: there is only one goal. The point of this example is to visually demonstrate the differing overall preference (graphed here as surfaces) over the design space, and to demonstrate that different design trade-off strategies can entirely change the solution.

A final observation about this example relates to computation. Equations 4.24 and 4.25 are misleading in their form. They are not unconstrained optimization equations. A simple unconstrained search algorithm on $\mathbb{R}^2$ will generally fail. Finding an initial point where the overall preference is non-zero is difficult, and constrained optimization techniques must be used to gradually introduce preferences which are initially zero. Preference function formulations do not change the computational complexity of the problem.

4.7 Related Work

Modeling decision making with real numbers is not new. There are many formalisms developed, and their use debated. Probability and Bayesian inferencing [21, 75, 176, 186], Dempster-Shafer theory [154, 159], fuzzy sets and triangular norms in general [42, 84, 85, 102, 105, 204], and finally utility theory [53, 58, 129] are all existing formal methods for representing uncertainty.

The similarities between these methods is that all them conform to the first three restrictions of Table 4.2 (boundary conditions, monotonicity, and continuity). Probability and Bayesian inferencing, Dempster-Shafer theory, fuzzy sets (triangular norms in general), and finally utility theory are all bounded between zero and one, and are continuous monotonic in the entries. The subsequent discussion, however, indicates where these theories diverge among themselves and with the development presented here. Each will be discussed at appropriate length, to demonstrate how each is related to the work. Then the discussion will relate optimization theory and matrix methods, as applied to design, to the development presented.

4.7.1 Utility Theory

The uncertainty representation of preference with the longest history is utility theory [53, 58, 129]. Utility theory was developed to assist in making a selection decision among a choice of actions, given uncertainty (noise) in outcomes of each action. Noise will not be discussed until Chapter 5, and so a comparison of this aspect will be deferred until then. However, a comparison of the axioms governing the imprecision aspect can be made.

Utility theory restricts to decision problems in which the individual preferences can be modeled as additive, either strictly, as expressed in Equation 4.13, or with a "multiplicative" form of

$$u(x^1, x^2) = k_1 u(x^1) + k_2 u(x^2) + k_3 u(x^1) u(x^2) \tag{4.26}$$

for a two goal problem, where the k_i are constants depending on the desires of the decision maker (roughly equivalent to importance) [58].

Since the overall metric is additive, utility theory reflects a compensating strategy, allowing the higher preference of some goals to compensate for the lower preference of others. Utility theory will therefore not admit non-compensating strategies.

Also, as discussed earlier, utility theory formulations, being additive, fail the annihilation restriction. To apply utility theory to design, this failure has been addressed by informally classifying the goals. Thurston has applied utility theory to design by dividing the complete set of goals into two classes: "objective" constraints and "subjective" goals [172, 173]. Objective constraints have crisp achievement levels that must be satisfied, and thus become standard constraints as in non-linear programming formulations. Subjective goals are those which can be traded-off, and are modeled using standard utility theory. Observe that this formulation results in exactly the characteristic function formulation as discussed in Section 4.5.3. That is, the overall preference metric is

$$\mu(d) = \prod_{j=1}^{m_g} \chi(g_j) * u(f^1, \ldots, f^{m_f}) \tag{4.27}$$

where f^i is a "subjective" goal, g_j is an "objective" constraint ($g_j > 0$ whenever the constraint is satisfied), $\chi(g)$ is 1 whenever $g > 0$ and 0 otherwise, and u is the utility function over the subjective goals. This formalism is difficult, as previously discussed, in that it is difficult to determine where to set the constraint levels on each "objective" constraint. How are these constraint levels traded off versus the utility of the "subjective" goals? The formalism of this chapter models this trade-off. Equation 4.27, however, does satisfy the restrictions of Table 4.2, and so if the designer can state the constraint values, it is a reasonable formalism for design. This is further supported by the belief that any formalism requires iteration. The "objective" constraint levels of the modified utility theory development (Equation 4.27) can be iterated over. This results in the "subjective" goals being traded-off among themselves in a compensating manner, and the "objective" constraints being traded off informally in the iteration. The overall strategy, therefore, remains informal, and not explicit to the designer.

This discussion has compared utility theory and the overall preference resolution (Definition 4.2) of the developed theory for design. Instead, a comparison between utility theory and the preference specification on each X_k can be made (Definition 4.1). Here the formalisms integrate. Each X_k consists of a set of inter-dependent parameters. Preference specifications on each X_k must be made considering each parameter (a d^i or p^j) within X_k. This is exactly as utility theory is intended to model. Thus, preferences on each X_k are determined using utility theory. Specification procedures and difficulties are discussed in Appendix A.

This makes clear the relationship between this development and utility theory. If the DPS $\times$ PPS reduces to a single X (*i.e.*, preferences can only be stated by considering all the goals simultaneously), then the development presented here reduces to utility theory.

4.7.2 Fuzzy Sets

Another formalism applied to representing imprecision in decision making is fuzzy sets [42, 84, 85, 102, 105, 204]. Fuzzy sets are intended to model subjective uncertainty for use in logic. Thus the intent is to construct subjectively uncertain version of "and" and "or" of classical logic. The fuzzy sets are derived from decision maker's linguistic, informal descriptions of desires, which are translated into formal ranks between zero and one. These are then combined to achieve a maximal decision (in the case of decision making) or to achieve an uncertain description of logical arguments (in uncertain logic more generally). The original presentation of fuzzy sets by Zadeh [200] combined uncertain information with the *min* function. Thus, Equation 4.5 is exactly the fuzzy set formulation of a design problem. This makes clear the relationship between this development and fuzzy sets. The development presented here reduces to using a fuzzy set resolution when trading off goals in a non-compensating manner, and without considering importance weightings.

The first paper describing the use of fuzzy sets for decision making was work by Bellman and Zadeh [15]. A decision was defined as a convolution of the constraints and goals, as done in this work. When such goals are equally important, then the non-compensating metric of Equation 4.5 was proposed as an overall metric. This work clarifies such a metric as adopting (informally) a non-compensating strategy. Bellman and Zadeh also realized that such a trade off was not always appropriate, and they suggested that at other times simple multiplication of the individual preferences might be appropriate. Using multiplication of preferences was seen as providing a "softer" combination. Simple multiplication is not the same as Equation 4.7, which normalizes the multiplication. This normalization is required for the idempotency restriction. Equation 4.7 is a true preference metric, whereas simple multiplication is not. Therefore a design rated with the *min* metric (Equation 4.5) can be formally compared (by looking at the overall metric result) with a design rated with the normalized product (Equation 4.7). This is not true of a comparison of any results from *min* and simple multiplication. Multiplication would be biased low.

The overall metrics of *min* and multiplication have been expanded to the more general class of *t-norm* functions, first proposed by Menger [94], whose various families are reviewed by Dubois and Prade [43]. T-norms are the set of functions $[0,1]^2 \rightarrow [0,1]$ bounded above by *min*, and serve as a model of uncertain conjunction. T-norms are thus not appropriate as an overall design metric, they are intended for logic. Probabilistic reasoning, Dempster-Shafer theory, and fuzzy sets are all t-norms [43].

Related to each t-norm is an associated *t-conorm* (or *s-norm*), which is the associated uncertain version of disjunction. T-conorms are bounded below by *max*. The set of functions between t-norms and t-conorms are the *mixed connectives*, bounded between *max* and *min*. This justifies the terminology for *IMC*, the set of idempotent mixed connectives. T-norms and t-conorms do have use with uncontrollable factors, as will be demonstrated in Chapter 5.

Using fuzzy sets for decision making has received much attention. Efstathiou and Rajkovič [51], Jain [77], Yager [194, 195, 196], and Zadeh [200, 202], for example, all discuss converting linguistic expressions into fuzzy sets, and then using the fuzzy mathematics to make decisions. Again, this represents using a non-compensating strategy for making decisions.

Importance criteria is also recognized in fuzzy set decision making. When importance information is available, Bellman and Zadeh suggested using a weighted sum of the preferences. Jain [78] and Bass and Kwakernaak [10] also develop additive metrics using fuzzy sets beyond Bellman and Zadeh's original work. Additive metrics have been discussed, and the reasons for not adopting them stated.

Yager [195] observed the "softness" of multiplication as a connective, and proposed it (and *min*) in conjunction with weights. Thus, Equation 4.8 was proposed for decision making by Yager. A non-normalized version of Equation 4.6, the non-compensating case, was also presented. Without the normalization, however, the non-compensating results are not directly comparable with compensating results.

Completely independent of fuzzy set formulations, Harrington [67] (in 1965, the same year as Zadeh's initial paper on fuzzy sets) proposed Equation 4.7 as a "desirability function" in chemical process problems. He as well observed the required annihilation condition for engineering design.

When the DPS $\simeq \mathbb{R}^n$ and the performance parameters are objective functions as in nonlinear programming, there has been much work with "fuzzy optimization." Here, preference functions are specified over the performance parameters, and most preferred design parameters searched for. Diaz [33, 34, 35] and Rao [132, 133, 134, 135, 136] have applied such fuzzy set formulations to design optimization problems in mechanical engineering. Nishiwaki [104] has applied fuzzy sets to nuclear radiation cover design. Again, this represents using a non-compensating strategy for making decisions. Fuzzy optimization is reviewed by Luhandjula [92].

Interactive methods with preference specifications has been pursued by Sakawa and Yano [143, 144, 145, 146, 147]. Here, preference satisfaction levels are specified on goals, and interactively updated. This is exactly the modified characteristic function formulation (Equation 4.11) as discussed previously. The trade-offs made among the satisfaction levels is not made explicit.

Other decision making methods have been developed using *fuzzy measures* [63, 74, 165], as originally developed by Sugeno [164]. Here one develops a fuzzy measure space, similar to a σ-algebra (also known as a *Borel field*), but where the measure is not required to satisfy additivity, only monotonicity. The fuzzy measure is interpreted as weight, and the space as the power set of the goals. Thus, larger subsets of the goals carry more weight than smaller ones. Further, since the measure is not assumed additive, unions of goals need not carry the weight of the sum of each goal's weight individually. Importance must therefore be specified on the power set of the goals, rather than on the goals (2^N specifications versus N at worst case). A decision is then developed as an integral across the fuzzy measure space, providing a weighted average metric.

This can be done, but the result is not guaranteed to be consistent with the restrictions of Table 4.2. Fuzzy integrals are integrals across a fuzzy measure space restricted to satisfy axioms of integration. The particular restriction of concern here is that if the integral is applied to a subset of the goals, all whose preferences are 1 (*i.e.*, the integral is applied to a characteristic function), then the result is the importance of the goals (the measure of the subset). This is similar to the integral of a characteristic function producing the size of the integration domain. This contradicts what is desired by a preference metric here. A combination of a subset of goals whose preferences are all 1 must result in 1, not the importance of the subset of the goals. Thus, fuzzy integrals fail the boundary condition of Table 4.2.

Fuzzy integrals form a class of integrals. Out of the class of fuzzy integrals, two in particular are discussed in the literature, Sugeno's integral and Choquet's integral. A further problem with these fuzzy integrals in particular is that both fail annihilation.

Thus, though conceptually the idea of a fuzzy integral as a rating of preference integrated over weight seems agreeable, the requirement of an integral to produce the weight when applied to a characteristic function negates its use here. The development presented in this chapter is intended for when the metric result is to be used as a design rating. A fuzzy integral fails this intention. It is intended to expand the concepts of integration to a more general structure than a σ-algebra. Fuzzy integrals can be used as a decision making formalism, but the restrictions that fuzzy integrals satisfy in such a domain must be scrutinized. They are inconsistent with the formalism desired here. Fuzzy integrals, however, will be used with uncontrollable factors, to be discussed in Chapter 5.

4.7.3 Optimization

Other methodologies exist that do not explicitly represent preferences on parameters. For example, simple optimization formulations (linear, non-linear, integer, and mixed integer programming) [3, 6, 123, 125, 137] assume a relationship between preference and the objective function: the lower the function, the higher the preference. Also, a relation is assumed between preference and the constraint functions: if the constraint is satisfied, the preference (for the constraint) is perfect. If the constraint just goes beyond its limit, instantly the designer does not prefer the design. Thus, using a single objective optimization formulation of nonlinear programming can be characterized by an overall preference metric of

$$\mu(\vec{d}) = \prod_{j=1}^{m_g} \chi(g_j) \frac{||f||_\infty - f(\vec{d})}{||f||_\infty} \tag{4.28}$$

where g_j is a constraint equation ($g > 0$ implies the constraint is satisfied), $\chi(g)$ is 1 whenever $g > 0$ and 0 otherwise, f is the objective function (assumed greater than zero), and $\vec{d}$ is a design parameter represented in vector form in some basis, reflecting the typical optimization problem.

The relationship between the development presented here and non-linear programming is that the development presented here reduces to single objective optimization formulations when the DPS and PPS are real valued, and the designer's preferences on the objectives are as above. The strategy exhibited by such a formulation is a non-compensating strategy. This is because, at any point in the DPS, one goal determines the preference. If all the constraints are satisfied, f determines the preference. If any constraint is not satisfied, that constraint dictates the preference as zero. Thus a non-compensating strategy (goal fixation) is exhibited by a single objective optimization formulation. The issue of weights is void, since any weight on any of the constraints or the objective function does not change its actions. Notice that the optimization formulation (Equation 4.28) satisfies the restrictions of Table 4.2.

Instead of a single objective formulation with constraints, others have proposed multi-objective optimization. Here, strategies are formally explicit only when a norm across the goals is used [148]. For example, weighted sum techniques [37, 52, 55, 106, 155, 160, 193] are compensating formulations: the higher performing objectives are averaged with the lower performing objectives. Weighted sums also usually incorporate importance weighting coefficients. As a specific example, consider "goal programming" [106, 149]. The Archimedian "goal programming" formulation is a weighted sum technique, with target values and nonlinear weights. Thus, it exhibits a compensating strategy. Additive metrics have been discussed in Section 4.5.3.

The pre-emptive "goal programming" formulation is a lexicographic order, with target values and nonlinear weights. Other lexicographic formulations can be found in the optimization literature [153, 155]. Lexicographic orders in general have been discussed in Section 4.5.3.

Minimax formulations [32, 106] are also non-compensating strategy formulations: they improve the lowest performing (highest valued) objective. Again, importance weighting coefficients can be incorporated.

Any multi-objective function norm minimization scheme can be characterized by an overall preference metric of

$$\mu(\vec{d}) = \prod_{j=1}^{m_g} \chi(g_j) \frac{||\vec{f}|| - ||\vec{f}(\vec{d})||}{||\vec{f}||} \tag{4.29}$$

where g_j is a constraint equation ($g_j > 0$ whenever the constraint is not violated), $\chi(g)$ is 1 whenever $g > 0$ and 0 otherwise, $\vec{f}$ is the vector of objective functions expressed in some basis, and $||\vec{f}||$ is the *sup* of $||\vec{f}(\vec{d})||$. Notice that any norm minimization scheme applied to multi-objective problem formulations require two decisions to be made. Both a goal resolving strategy (norm) and goal weighting coefficients (possibly identity) must be chosen. These decisions are independent of each other. The norm minimization formulation

(Equation 4.29) only satisfies the restrictions of Table 4.2 when the norm is the infinity norm (*sup*), thus only when exhibiting a non-compensating design strategy. A two norm (addition) fails the annihilation condition, and thus is not the best formulation for design (this is more generally true for any p-norm of finite p, as in Equation 4.15). Biegel and Pecht [18] or Vincent [181], for example, also make this argument for the 2-norm.

Typically, multi-objective formulations are used iteratively, without specifying an explicit strategy. Such methods have been used in design [8, 130]. Also, there are formal algorithms for such iteration. STEM [16], GDF [60], or the VI algorithm [86], for example, interactively question a decision maker about relative trade-off preferences. This induces a locally relevant set of weights, and the algorithm proceeds. Such algorithms are based on an un-expressed overall metric consistent with utility theory (see Steuer [160]), and so these multi-objective algorithms exhibit a compensating design strategy. They can be considered as using an additive metric (perhaps non-linear) without explicitly expressing the metric. Additive metrics and utility theory have been discussed in Sections 4.5.3 and 4.7.1.

In the situation of the DPS $\simeq \mathbb{R}^n$, optimization methods differ with this development in the formalization of the objectives: *what* should be optimized. It must be made clear, however, that the development here is as difficult to evaluate as non-linear programming formulations, in general. Definition 4.4 defines a constrained optimization problem: maximize μ subject to $\vec{d}$ being within the support of μ. Ensuring that this constraint is satisfied is non-trivial, and will, except in the easiest of problems, require nonlinear programming formulations such as a boundary method [123] to gradually incorporate preferences which are initially zero.

4.7.4 Matrix Methods

Another method with historical development in engineering design is the use of morphological charts [4, 5, 9, 18, 126]. When using a formal chart, alternatives are listed versus evaluation criteria. Each alternative is ranked on each criteria, and the alternative with the best aggregated criteria is used. This is the general problem presented in Example 1, Section 4.6.1. Thus, the morphological analysis methods can be placed on a sound theoretical foundation by converting the ratings to preferential ranks, and using the machinery developed in this chapter.

Morphological analysis has not been looked on favorably by some (see Pugh [126]), since the analysis selects alternatives early, and subsequently the selected alternative may exhibit difficulty [126]. Pugh therefore suggests that weighted sums not be used, and instead the designer should make a selection by observing the ranks carefully, and not always make a judgment based on additive sum. He presents a technique of summing the negative and positive aspects of each alternative as well, and then making an informal decision based on these ranks. Such procedures involve the designer ranking the designs informally. The work presented in this section, which makes explicit that different strategies may be appropriate, can return formalism to such a selection procedure. The designer could use various metrics, and observe the various results, and make detailed formal analyses at this preliminary stage, as Pugh suggests, rather than simply accepting a weighted sum solution as "correct."

Such different methods have been analyzed by others. In a classic paper, Milnor [96] considers the choice among four alternatives, and demonstrates how four different strategies selected each of the four different alternatives. This is not seen as a refutation of the formalization; rather, it indicates that a designer must determine how to trade off among goals, and that doing so can entirely change the outcome.

A well developed chart based method is the "Analytic Hierarchy Process," or AHP, originally developed by Saaty [62, 140, 141]. This is a chart based method for determining relationships between alternatives: each alternative is listed on each axis of the chart. Then a relative "distance" (based on some criteria) is established from pairwise comparisons. For example, the "distance" could be relative importance. With suitable assumptions, the eigen vector of the maximum eigen value is the normalized distance among the alternatives (the normalized importance weights, for example).

AHP has been used for selection among alternatives. Based on criteria, each alternative is compared relative to all the other alternatives. The eigen vector of the greatest eigen value of the comparison matrix then becomes the normalized ratings of each alternative. Thus, the largest component in the eigen vector corresponds to the highest rated solution.

AHP also has an axiomatic foundation (see Saaty [142]), and a restriction required to use AHP is that an Archimedian property must be satisfied. Thus the annihilation condition is not satisfied by AHP. Further, AHP constructs the rank using a ratio scale, which means the scale is invariant under similarity transformations. This means the rank is derived from a linearly additive preference function (see French [58] for a more complete discussion). Additive combinations of preferences have been discussed in Section 4.5.3. AHP can, however, be used to assess values for the importance weightings used in the development presented here: importance is a relative measure.

A final matrix method that will be discussed is the formal chart devices used in "Quality Function Deployment," or QFD [4, 69]. Here, a device is broken down into components, which are listed versus the goals of the design. Each goal is then matched to a component, to ensure each goal is satisfied. The matching is represented with a formal symbol from the set $\{\circledcirc, \bigcirc, \triangle\}$, formalizing greatest to least satisfaction. Ratings are performed by using a transformation to convert the symbols to numerical equivalents, and summing. This summation has the shortcomings of the additive metrics as discussed in Section 4.5.3. Clearly, the summation procedure can be replaced by the methodology of this chapter, forming a sound theoretical basis for QFD. For example, a non-compensating combination could be performed if some aspects of the design must be ensured, rather than always adopting a compensating strategy.

4.8 Conclusions

Imprecision is an uncertainty form in design that has many possible candidate formalisms. The formalism developed here is believed comprehensive, though indeed no such belief can be proven: all formalisms are developed from informal characterizations, and informal statements cannot be proven.

The comprehensiveness is reflected in the set IMC. This set provides a framework for analyzing new and existing design technologies as they are developed. Any metrics proposed for design can be compared with the restrictions defining IMC, and evaluated. Thus, the developed axiomatization of preference also has application beyond preference based methods. It forms the framework by which other, non-preference methods can be judged. For example, optimization methods were clarified in Section 4.7.3. This can be repeated for any new methodology, to understand the restrictions the methodology conforms to, and the implications of such conformance.

Selecting any particular metric in design is an informal process. How goals are traded off is a determination of designer intention. Non-compensating and compensating are informal notions. Since intention is informal, it can have infinite formalizations. Thus determination of any formal metric must be scrutinized based on the restrictions it satisfies, in informal terms. The principle metrics proposed are the non-compensating and compensating metrics of Equations 4.5 and 4.7. These can be convolved for more complex metrics. Other commonly used metrics were also compared, since they have been used in design. Use of any metric should be accompanied with an understanding of the restrictions the metric obeys, and the restrictions the metric does not.

The non-compensating metric of Equation 4.5 has been rejected by others since it is not guaranteed to produce a solution in the Pareto optimal set [57]. The metric can select dominated solutions along with non-dominated solutions. This can occur when the lowest performing goal has equally low preference between two solutions, but the remaining goals have different preference. Notice this does not mean, as some interpret, that the metric will select a dominated solution, and no other. It may select a dominated solution in addition to non-dominated solutions, and make no further determination. The non-compensating metric does not differentiate such designs. The metric performed as desired, however. It determined all d^* with supremum of the lowest performing goal. This calculation will present all such solutions, and the designer can subsequently make a formal choice over this restricted set.

The developed theory expressed axioms in terms of the model: $\mathcal{P}$ combines μ_k, and so the restrictions on $\mathcal{P}$ are discussed in terms of μ_k, rather than on a more "basic" formal order relation over the DPS $\times$ PPS. This work presented in this chapter directly converted informal arguments into formal restrictions on $\mathcal{P}$. It did not convert informal arguments into formal restrictions on an ordering over DPS $\times$ PPS, and then formally derive the restrictions on $\mathcal{P}$ by theorems based on the determined order relation restrictions. French [58], for example, views this as a theoretical weakness, since he instead feels any restrictions should be stated in terms of how the designer makes preferential decisions, as represented by an ordering over the alternatives. The order relation formalism is desired, since the order relation is assumed to "be" the designer's informal desires. The restrictions in Table 4.2 are not in terms of a formal order relation.

The argument is muted, however, when the distinction between informal and formal models is understood. An order relation is not an informal concept, it is a formal representation of the designer's desires. Thus, the informal to formal model translation occurs in either case, and in either case the restrictions must be interpreted informally. Whether the restrictions are expressed in an order relation formalism or in a combination function formalism is seen as irrelevant. The order relation is only a formal reflection of the informal characterization, it is not the informal characterization.

A key problem with the developed methodology is the determination of the individual preferences μ_k. The procedures developed by utility theory can be used for this purpose. It has been shown, however, that even these well developed methods suffer from inconsistencies [13, 70, 158, 178]. How the preferences are elicited varies the preferences. This simply reinforces the idea that iteration is required in design. Any preference specification can only be considered preliminary. It should be varied, and the results observed. These formal observations should then be compared by designers with their informal desires; and any discrepancies will result in modifications. This iteration involves the informal model. Thus, it is not believed that any such iteration can be formalized into an algorithm. Desires are informal concepts. The formal model can be iterated to a formal solution, but it must still be interpreted informally.

The need for informal-formal iteration is also reinforced by the understanding that some of the specified preferences usually arise from customer desires. In general, however, this does not mean engineering design is a group decision making enterprise. As preference for such goals, designers adopt their understanding of what the customer desires; they do not negotiate with the customers as they perform the design, and some of the designer's goals are non-negotiable. Thus, problems of group decision making are avoided [7, 58] in this development. If the design determined with the initial customer specifications are, however, less than satisfactory, then the customer preferences may also need revision. This revision of the individual preference is a group negotiation. Once such individual preferences are formalized, then the problem returns to having a single source of preference, which the developed formalism operates with. Thus the group aspect of designer and customer decision making occurs in any negotiation during the formulation of such preferences, not in the combination of such preferences by the designer with other preferences. This clarification agrees with previous characterizations of engineering design processes [4, 5, 9, 69].

It is believed a reasonable defense of the proposed restrictions and methodology has been made. In any case, it not assumed that any development should be accepted normatively, and applied. Rather, the developments form a descriptive model of what can be used: each development can be applied, and results observed. A designer can pursue "what if" questions: if a non-compensating approach is taken, what are the results; or if a compensating approach is taken, what are the results. Further, the formalism described ensures the results are always formally directly comparable. Only after such iteration and informed comparisons should decisions be made.

Chapter 5

Designs with Uncontrollable Parameters

This chapter will present a formal methodology for modeling and solving design problems with noise, as originally developed in [112]. Examples of noise in engineering design are: manufacturing uncertainties and tolerances, variations in material properties, or variations in the decisions of others in a design process. Noise will be explicitly modeled, and mathematics for manipulating noise will be developed. Specific examples will then be presented using the developed formalisms.

5.1 Introduction

This chapter assumes a formal design parameter space, DPS, characterized by design parameters d_i, and a performance parameter space, PPS, characterized by performance parameters p_j. This is further assumed complete with preference specifications μ_k as defined in Definition 4.1, and a formal objective resolving strategy $\mathcal{P}$ as defined in Definition 4.2. Thus all the development of Chapter 4 will apply. Also, since Chapter 4 presented the simplifications required of the development to reduce to other formal engineering design methods, this chapter applies to them equally well.

This chapter now considers the effects of a noise parameter space, NPS. Typically in an engineering design problem, noise is characterized by *noise parameters*, $n_1, \ldots, n_k, \ldots, n_q$. A noise parameter n_k might be the possible positioning of an operator switch, and so the alternatives may be finite. Alternatively, n_k might be a value of a manufacturing error on a design parameter, and so the NPS may have a continuum of possibilities. In this case,

n_k might be thought of as a vector within an NPS $\simeq \mathbb{R}^q$ represented with coordinates n^t in some basis, for example.

Though these specific, simple examples illustrate the concept of noise, the structure needed to model noise can be developed more generally. Thus, the NPS is defined to be an *uncertainty measure space*, or a set with additional measure structure, to be defined.

Definition 5.1 *An* uncertainty measure space $(\text{NPS}, \mathcal{B}, \text{g})$, *is a set NPS of elements n, a σ-algebra $\mathcal{B}$ of sets over the NPS, and an uncertainty measure $g : \mathcal{B} \rightarrow [0, 1]$.*

An uncertainty measure is different from a Lebesgue measure [64] or a fuzzy measure [43, 50, 164, 165]. An uncertainty measure is a function $g : \mathcal{B} \rightarrow [0, 1]$ intended to measure the effects of noise. Three specific measures will be developed: the probability measure Pr, the possibility measure Π, and a necessity measure $\mathfrak{N}_\alpha$.

In keeping with the terminology of probability, an element of $\mathcal{B}$ is called an *event*. The measure g is to be interpreted as a formalization of the ability of an event to occur.

The σ-algebra $\mathcal{B}$ (also called a *Borel field*) is determined by the designer. $\mathcal{B}$ characterizes the ability of the designer to make statements about the NPS. The number of subsets within $\mathcal{B}$ is determined by the designer's ability to characterize the NPS.

That events in the NPS have the structure of a σ-algebra must be justified. Formally, this means the NPS has an associated collection of subsets $\mathcal{B}$ that satisfy, $\forall\ N_i,\ N_j \in \mathcal{B}$:

$$\begin{aligned} &i) \quad \neg N_j = \text{NPS} \backslash N_j \in \mathcal{B} \\ &ii) \quad \cup_{j \in J} N_j \in \mathcal{B} \end{aligned} \tag{5.1}$$

where J is an index set. Thus, if an event can occur, it can also not occur. If 2 events can occur separately, then either event can occur. This is true for any sequence of events. DeMorgan's laws also hold on subsets:

$$\begin{aligned} &iii) \quad \neg(N_i \cup N_j) = \neg(N_i) \cap \neg(N_j) \\ &iv) \quad \neg(N_i \cap N_j) = \neg(N_i) \cup \neg(N_j). \end{aligned} \tag{5.2}$$

Thus, if either of two events does not occur, then both do not occur. Also, if neither event can occur together, then either one or the other cannot occur.

These assumptions are sufficient to show the collection $\mathcal{B}$ forms an algebra over the subsets [64]. Thus, events in $\mathcal{B}$ also satisfy:

$$\begin{aligned} ii)&\ \ \cup_{j\in J} N_j \in \mathcal{B} \\ v)&\ \ \cap_{j\in J} N_j \in \mathcal{B} \end{aligned} \tag{5.3}$$

where J is an index set. The algebraic structure of a noise space is not new to this work, it is historically well developed [64] (for probability).

Given an uncertain space (NPS,$\mathcal{B}$,g), the NPS is characterized. What is desired is to rate a design configuration, given these noise effects. To do so, a disjoint collection of subsets whose union is the entire NPS, or a *partition*, is needed. On each subset in the partition, the effects of the noise will be determined, and then each rating on each subset in the partition will be incorporated into an overall rating across the partition (across the NPS). Not just any partition is used, but the limit in refinement of any sequence of partitions within $\mathcal{B}$. Thus, the most accurate rating of the noise is used, given what the designer can state about the noise.

The effects of including noise will now be considered. Any changes in the solutions of Chapter 4 due to the addition of noise will be formally developed.

5.1.1 Measuring the Effects of Noise

For engineering design, a NPS is introduced to ensure that the design will perform in the various noise parameter configurations that are possible. The NPS itself provides a formalization of what effects can occur, and the measure provides a formalization of the degree. The rating of these effects remains to be determined, given their degree of ability to occur.

That is, the NPS models what the designer does not have a choice over within the formal model. Each element within the NPS has an associated measure of ability to occur within the model. The effects of the noise must be modeled with respect to the measure of the ability of the noise to occur. Thus, each smallest identified disjoint subset N of the NPS is incorporated into a rating of a fixed element d of the DPS, with respect to the measure of the noise N's ability to occur. This implies an *integral* of the performance across the

NPS with respect to the measure of noise should be used. Each element in the NPS is incorporated, to the degree specified by the measure, into the overall rating of a design parameter set. Formally, this will be done by introducing an *uncertainty integral* across the NPS.

Before this can be done, it must be clarified what can be measured. All performance parameters and preferences are assumed to be Lebesgue measurable [139] across the NPS with fixed design parameter values d. Thus, the performance parameters and the σ-algebra $\mathcal{B}$ are well formed.

Definition 5.2 *An* integral *of a function* f :NPS $\rightarrow [0,1]$ *with respect to an uncertainty measure* $g : \mathcal{B} \rightarrow [0,1]$ *over an uncertainty space* $(\mathrm{NPS}, \mathcal{B}, g)$*, denoted*

$$\int_N f(n)\, dg : (\mathrm{NPS}, \mathcal{B}, g) \rightarrow [0,1]$$

is a map used to determine the performance rank of the function f *over a subset* N *of the NPS as measured by the uncertainty measure* g*.*

This definition is quite general. For design purposes, there are specific restrictions the operation must satisfy, delineated here.

The first restriction is that the term "integral" must be justified. The uncertain integral of a characteristic function must produce the uncertain measure of the set:

$$\int_{NPS} \chi_N(n)\, dg = g(N). \tag{5.4}$$

Thus the integral of the characteristic function of a subset, which indicates the subset, should result in the ability of the subset to occur.

Second, the uncertain integral of a constant function must produce the constant.

$$\int_{NPS} \alpha\, dg = \alpha. \tag{5.5}$$

Thus, if the same performance results no matter what element in the NPS occurs, this level of performance should be the performance rating.

The uncertain integral must be monotone in the argument:

$$f_1(n) \geq f_2(n) \Rightarrow \int_N f_1(n)\, dg \geq \int_N f_2(n)\, dg. \tag{5.6}$$

Thus, higher performance at each element within the NPS results in a higher performance rating.

Finally, the uncertain integral must be monotone in inclusion for all N_j, N_k:

$$\begin{gathered} N_j \supseteq N_k \Rightarrow \textstyle\int_{N_j} f(n)\, dg \geq \int_{N_k} f(n)\, dg \\ \text{or} \\ N_j \supseteq N_k \Rightarrow \textstyle\int_{N_j} f(n)\, dg \leq \int_{N_k} f(n)\, dg. \end{gathered} \tag{5.7}$$

This statement indicates one or the other holds for all N_j, N_k. Thus, as more of the NPS is considered, there is more ability for the noise to affect the performance rating.

Any operation on an NPS that satisfies these conditions will be called an *uncertainty integral*.

Definition 5.3 *The set of* uncertainty integrals, *denoted* UI(NPS, $\mathcal{B}$, g)*, is the set of all integrals satisfying Equations 5.4 through 5.7.*

This set provides the framework for calculating with an NPS.

5.1.2 Design Calculations with Noise

The framework for evaluating performances across the NPS has been established. The performance to be measured across the NPS (f in Definition 5.2), in the context of selecting most preferred design parameters (as in Chapter 4), is the overall preference $\mu(d, n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$. Thus, an element $d \in$ DPS can be rated using an operation satisfying the uncertainty integral restrictions:

$$\mu(d) = \int_{NPS} \mu(d, n)\, dg \tag{5.8}$$

$\mu(d)$ as above is therefore the *expected* preference given the NPS. Further, the n are those used given a fixed d. The NPS can generally vary with d.

To clarify the development at this point: it is clear that Equation 5.8 can be satisfied by many operations. UI contains many operations, the uncertainty measure g can be satisfied by many different measures, and $\mu(d,n)$ can be determined by many different $\mathcal{P}$. The uncertainty integral will be made well defined for specific noise forms (specific measures and integration operations will be developed), and $\mathcal{P}$ can be made well defined using the development of Chapter 4 (specification of a strategy).

Equation 5.8 reflects the *expected overall preference*. Given the NPS, this metric provides the expected overall preference as formalized by the designer. This is not the only possible technique. One could as well define:

$$\mu(d) = \mathcal{P}\left(\int_{NPS} \mu_1(d,n)\, dg, \ldots, \int_{NPS} \mu_N(d,n)\, dg\right). \tag{5.9}$$

This reflects an overall combination of the expected individual preferences. This does not, however, consider that $\mathcal{P}$ may change due to the NPS. The strategy $\mathcal{P}$ may switch due to a performance parameter value changing with different values of n. Equation 5.9 assumes that the strategy is constant across the NPS.

Having developed a formalism that rates the effects of noise, the formal problem for designs with uncontrollable factors is now framed. There are goals whose preference must be maximized across a design space, but whose evaluation is confounded by noise. There is also, however, a measure of this confounding influence. With noise, the net overall performance is determined by integrating the measure of the specific performances across the noise space. Thus, the performance improvement principle (Assumption 2.1), as applied in this case, formalizes into maximizing the integral of preference across the noise. Formally,

Definition 5.4 *The most preferred points* $\{d^*\} \subseteq \mathrm{DPS}$ *are defined by*

$$d^* : \mu(d^*) = \sup\left\{ \int_{NPS} \mu(d,n)\, dg \;\middle|\; n \in NPS,\ d \in \mathrm{DPS} \right\}$$

where

$$\mu(d,n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$$

is the overall preference calculated at a point $d \in DPS$ *with the noise fixed at* $n \in NPS$*, and*

(NPS, $\mathcal{B}, g$) *is an uncertainty space.*

Thus, $\mu(d^*)$ is the maximum *expected* preference. Again, generally the NPS can vary with d. The NPS, $\mathcal{B}$, and g are all dependent on d.

Given that a measurement of noise must be made, there are many different methods to perform the measurement and integration. Different types of noise require different measures. The different measures that will be developed directly correspond with the different uncertainty forms described in Chapter 3. These distinctions and associated formalizations will now be developed. First, however, the relation between commonly formalized noise parameters and the σ-algebra will be clarified. Then Example 3 will be expanded to include different effects of noise.

5.1.3 Noise Parameters

For the formal definitions, the NPS is only required to be a set NPS with an associated σ-algebra structure. Typical engineering examples, however, have more structure than this, and are usually described with *noise parameters*. This section will discuss the relation between noise parameters n_k and the events N.

Problems described with noise parameters typically have more structure than that required for the definitions. The examples considered here are: the discrete case when the NPS $\simeq \mathbb{Z}_q$, the continuum case when the NPS $\simeq \mathbb{R}$ locally, and when the NPS is equivalent to Cartesian products of these locally.

In the discrete case of NPS $\simeq \mathbb{Z}_q$, the natural collection of subsets

$$2^{\mathrm{NPS}} = \{\emptyset, \{n_1\}, \ldots, \{n_q\}, \ldots, \mathrm{NPS}\} \tag{5.10}$$

of the power set forms a natural σ-algebra to use.

In the continuum case of NPS $\simeq \mathbb{R}$, there is the collection of all open, half open, and closed intervals of $\mathbb{R}$. A natural σ-algebra to use is the smallest σ-algebra containing the usual topology of $\mathbb{R}$ (unions all open intervals), the collection of which is known as the *Borel sets*,

$$\mathcal{B}_B = \bigcup_{j \in J} [a_j, b_j) \quad \text{where } a < b \tag{5.11}$$

which are the unions of all half open (on the right) intervals of $\mathbb{R}$, $[a, b)$. These form the natural subsets to use in the definition.

Obviously, these cases can be combined in ordered pairs for a Cartesian NPS. Further, this need only hold locally. The noise parameters can be local parameterizations of an NPS with manifold structure, for example. For example, a revolute joint in a design can be naturally modeled as $\mathbb{S}^1$, the unit sphere lying in $\mathbb{R}^2$. It may be subject to random disturbances. This can be formalized into a design parameter angle $\theta \in [0, 2\pi) \subset \mathbb{R}$, with a variation $\delta\theta$ reflecting random variation from the reference position. θ is a particular parameterization of the DPS $\mathbb{S}^1$, and $\delta\theta$ is a particular parameterization of the NPS $\mathbb{S}^1$. The stereographic projection is another equally valid parameterization of $\mathbb{S}^1$ using $x \in \mathbb{R}$, with a different δx parameterizing the NPS $\mathbb{S}^1$. These are formalization process concerns.

Two typical examples of an NPS ($\mathbb{Z}_q$ and $\mathbb{R}$) were described in this section. Each will be specifically shown for each noise form, as each noise form is formalized.

5.2 Probabilistic Noise

A particular uncertainty form that can be used to model the NPS occurs when the events are random, as informally defined in Definition 3.5. For example, inaccuracies in measurements and manufacturing are usually modeled as random. Such inaccuracies form what is now termed a *probability space*. The probability space will be denoted NPS, meaning all uncertainties in the NPS are now considered probabilistic, for this section. Any given inaccuracy within is termed an event, and events are assumed to obey the restrictions of a σ-algebra. Thus, if an event can occur, it can also not occur. If 2 events can occur separately, then both events can occur. This is true for any sequence of events. Again, the algebraic structure of a probability space is not new [87, 168].

Given the probability space, an uncertainty measure g is constructed, and denoted Pr. Pr measures the probability of an event occurring. A restriction of the Pr measure is that the probability of an event occurring and not occurring must equal the probability of the certainty event (assumed to be 1) under real addition. Further, the probability of either of two disjoint events occurring must equal the real additive probability of the two events. These restrictions are sufficient to derive an uncertainty measure Pr [28, 42, 43, 87, 176].

Thus, for events N_j, $N_k \in \mathcal{B}$ such that $N_j \cap N_k = \emptyset$, if g is restricted to obey:

$$\begin{aligned} g(N_j) + g(\text{NPS} \backslash N_j) &= 1 \\ g(N_j) + g(N_k) &= g(N_j \cup N_k) \end{aligned} \tag{5.12}$$

then g is a probability measure of classical probability theory [28, 87, 176].

Definition 5.5 *Any uncertainty measure g satisfying Equations 5.12 will be called a* probability measure, *and denoted* Pr.

The probability measure is a Lebesgue measure of Lebesgue integration theory [65], restricted such that the measure of the whole space is 1.

This formalism now allows the measurement of the effects of noise on the performance, given any fixed design parameter arrangement d. Across the NPS, a disjoint collection of subsets $N \in \mathcal{B}$ whose union is the whole NPS is used (a partition of the NPS). The effect of each noise event N can be accounted for by determining the performance μ at any point $n \in N$, and weighting it by the probability of N occurring. This can then be summed across the partition. The limit as the partition becomes finer in $\mathcal{B}$ is used as the overall rating for d.

Thus, performance at any point d in the DPS now becomes the integral of performance across the probability space.

Definition 5.6 *Given a probability space* (NPS, $\mathcal{B}$, Pr)*, the preferential performance of a point $d \in$ DPS is defined by*

$$\mu(d) = \int_{NPS} \mu(d, n)\, dPr$$

where $\mu(d, n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$.

This integral is the standard Lebesgue integral from measure theory [64, 139]. Thus, $\mu(d)$ is the probabilistic expectation of $\mu(d, n)$ across the probability space with respect to the probability measure Pr.

Now an expected preference is determined for each design parameter $d \in$ DPS, given the uncontrollable NPS. A performance rating has been obtained. Thus, the performance improvement principle (Assumption 2.1) can be invoked, and the highest performing design parameters defined. With the specific uncertainty measure of probability, Definition 5.4 implies the most preferred points $\{d^*\} \subseteq$ DPS are defined by

$$d^* : \mu(d^*) = \sup \left\{ \int_{NPS} \mu(d,n)\, dPr \;\middle|\; n \in \text{NPS},\ d \in \text{DPS} \right\} \tag{5.13}$$

where

$$\mu(d,n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$$

is the preference calculated at a point $d \in$ DPS with the noise fixed at $n \in$ NPS.

5.2.1 Probabilistic Noise Parameters

In this section, the probability space will be clarified when particular noise parameters are used within a formal model. The discrete case of NPS $\simeq \mathbb{Z}_q$ and the continuum case of NPS $\simeq \mathbb{R}$ will be made explicit.

Consider a discrete NPS $\simeq \mathbb{Z}_q$. Then each of the individual events $\{n_j\}$ forms a suitable partition, where each has an associated discrete $Pr(\{n_j\})$. The integral of Definition 5.6 then becomes

$$\mu(d) = \sum_{j=1}^{q} \mu(d, n_j) \cdot Pr(\{n_j\}|d). \tag{5.14}$$

Consider NPS $\simeq \mathbb{R}$. Then each of the individual Borel sets $[a,b)$ (as discussed in Section 5.1.3) has an associated $Pr([a,b))$, and a partition of the form $[n, n+dn)$ might be used. This can usually be characterized by a probability density function $pdf(n)$. The integral of Definition 5.6 then becomes

$$\mu(d) = \int_{NPS|d} \mu(d,n)\, pdf(n|d)\, d(n|d) \tag{5.15}$$

a Riemann integral over the NPS (which can, as before, vary with d). Of course, the PPS, preferences, and density functions must all be Riemann integrable for this to hold.

As an example, consider a one parameter design, with a preference $\mu(d)$ and a $pdf(\delta d)$ probability density as graphed in Figure 5-1. The $pdf(\delta d)$ is the same for all design parameter values: the NPS does not change with d. The maximal *expected* preference, *i.e.*, the maximum of the $E[\mu]$ function (which is the result of applying Equation 5.15), not the μ

function maximum, is found. This set $\{d^*\}$ is shown on the design parameter axis.

For designs with no probabilistic uncertainty, the parameters in the design model which are not design parameters are all crisp numbers. In such circumstances, Equation 5.15 reduces to Definition 4.4, since the probability density functions reduce to delta functions at the crisp values.

5.2.2 Discussion

Definition 5.6 is the standard first moment of expectation [168] of the preference μ. That is, Definition 5.6 represents the average performance μ across the noise. Clearly higher moments of this relation can be obtained, to derive the variance, skew, and kurtosis, for example. Clearly this additional information can aid the designer.

Care should be taken with probabilistic spaces, however. If the development outlined in this section is taken, it must be understood that one is designing to improve the *average* performance. The NPS must be examined to ensure this is appropriate.

For example, consider an NPS $\simeq \mathbb{R}$ with a split probability density that results in performance as graphed in Figure 5-2. In this case, improving the average performance may prove difficult, since the average is impossible.

5.3 Possibility

Possibility can be used to represent parameters within a formal model the designer does not have choice over, and that are not characterized by probability. Subjective choices of others (not the designer), for example, can be modeled with possibility.

Definition 3.6 in Chapter 3 informally defined possibility as uncertainty in the limits of capacity within a formal model. Thus, a possibilistic variable can have a range of values, but the range is limited by another person's choices. The range of another person's (other than the designer's) choice is an example of what is meant by a capacity.

Similarly to the previous form of probability, the uncertainties defined in Definition 3.6 will be formalized into what is now termed a *possibility space*. The possibility space will be denoted NPS, meaning all uncertainties in the NPS are now considered possibilistic, for

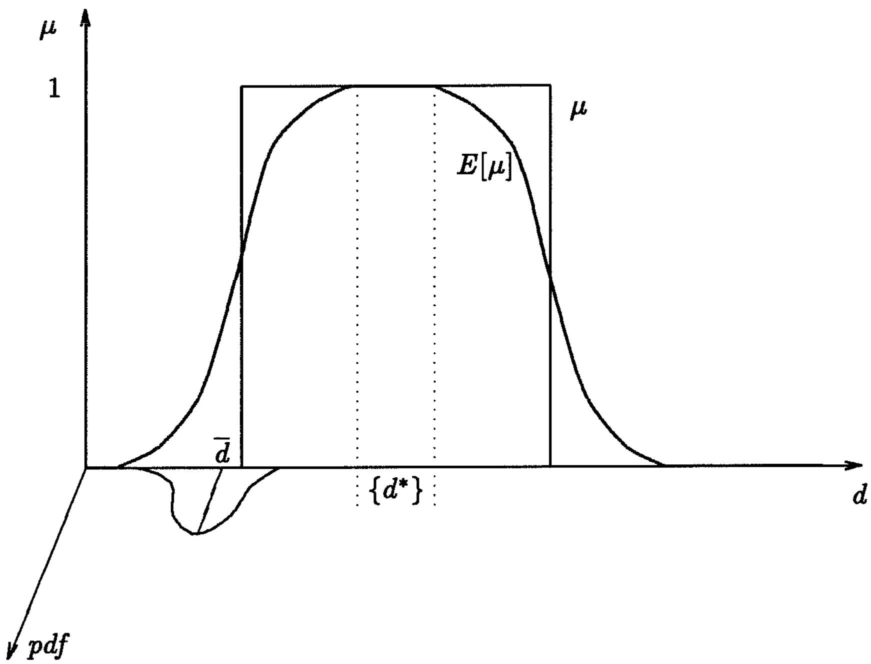


Figure 5-1: Parameter resolution with probabilistic uncertainty.

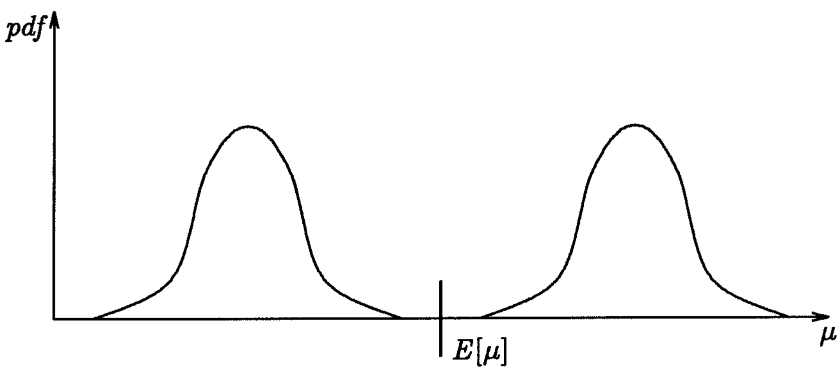


Figure 5-2: Expected value maximization counter-example.

this section.

Possibilistic uncertainty can formally modeled as a continuous *fuzzy measure*, as originally developed by Sugeno [164]. A fuzzy measure [43, 50, 164, 165] is a function $g : \mathcal{B} \rightarrow [0,1]$ such that, for $N_j, N_k \in \mathcal{B}$:

$$\begin{aligned} &i) \quad g(\text{NPS}) = 1 \quad g(\emptyset) = 0 \\ &ii) \quad N_j \subseteq N_k \Rightarrow g(N_j) \leq g(N_k). \end{aligned} \tag{5.16}$$

Continuity can also be required:

$$iii) \quad \lim_{j\to\infty} g(N_j) = g(\lim_{j\to\infty} N_j) \quad \text{for nested sets } N_j. \tag{5.17}$$

A fuzzy measure is not the same as a Lebesgue measure [64], which replaces conditions *ii)* and *iii)* with additivity conditions. Clearly, however, the probability measure is a fuzzy measure (additivity implies monotonicity).

Uncertainty in capacity (as defined by Definition 3.6) can be modeled as a fuzzy measure. The first restriction is justified in that, if there is capacity in the uncertainty model, then the possibility of a value to use must be 1. Similarly, the possibility of the value to use must be 0 with a capacity of nothing. Increasing the possibility per event must increase the possibility. Increasing the event size must increase the possibility.

For all events, either the event or not the event is possible. If two events are possible, a choice must still be made (but not by the designer), since only one event can occur. So if two events can occur, it is assumed the other agent will as well choose to maximize performance, and so the capacity of using either of two events will be the greater of the two, since this option would be chosen. Therefore, for possibilistic events $N_j, N_k \in \mathcal{B}$:

$$\begin{aligned} \max\{g(N_j), g(\text{NPS} \backslash N_j)\} &= 1 \\ g(N_j \cup N_k) &= \max\{g(N_j), g(N_k)\}. \end{aligned} \tag{5.18}$$

Equations 5.18 defines a possibility measure [50].

Definition 5.7 *Any uncertainty measure g satisfying Equation 5.18 will be called a* possibility measure, *and denoted* Π.

This formalism allows the measurement of the effects of noise on the performance, given any fixed design parameter arrangement d. Across the NPS, a disjoint collection of subsets $N \in \mathcal{B}$ whose union is the whole NPS is used (a partition of the NPS). The effect of each possible event N can be accounted for by determining the performance μ at a point $n \in N$, and then ensuring $\Pi(N)$ is within capacity at this evaluation point. Since the design is limited by the capacity Π, the design can be rated no better than the capacity Π. After this attenuation, the best possibility in the NPS can be used.

Thus, the integral of performance across the possibility space becomes the maximum possible performance, with performance attenuated to be possible to the degree specified by the possibility measure.

Definition 5.8 *Given a possibility space* (NPS, $\mathcal{B}$, Π), *the preferential performance of a point* $d \in \text{DPS}$ *is*

$$\mu(d) = \sup\{\min\{\mu(d,n), \Pi(N)\} \mid N \in \{N_j\} \textit{ disjoint} \subset \mathcal{B}\}$$

where $\mu(d,n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$, $n \in N$.

This integral is the Sugeno integral of possibility theory [164], expressed for when the NPS has the structure of a σ-algebra. Note the *sup* is across the subsets of the partition $\{N_j\}$, and the limit as the partition becomes finer in $\mathcal{B}$ is used. Thus, $\mu(d)$ is the possibilistic expectation of $\mu(d,n)$ across the possibilistic noise space with respect to the possibility measure Π.

Now an expected preference is determined for each design parameter $d \in \text{DPS}$, given the uncontrollable NPS. A performance rating has been obtained. Thus, the performance improvement principle (Assumption 2.1) can be invoked, and the highest performing design parameters defined. With the specific uncertainty measure of probability, Definition 5.4 implies the most preferred points $\{d^*\} \subseteq \text{DPS}$ are defined by

$$d^* : \mu(d^*) = \sup\{\sup\{\min\{\mu(d,n), \Pi(N)\} \mid N \in \{N_j\} \text{ disjoint} \subset \mathcal{B}\} \mid d \in DPS\} \quad (5.19)$$

where

$$\mu(d, n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$$

is the preference for a point $d \in$ DPS, given the noise fixed at $n \in N$. $\mu(d^*)$ is the overall expected preference for the most preferred solution.

5.3.1 Possibilistic Noise Parameters

In this section, the possibility space will be clarified when particular noise parameters are used within a formal model. The case of NPS $\simeq \mathbb{Z}_q$ and NPS $\simeq \mathbb{R}$ will be made explicit.

Consider NPS $\simeq \mathbb{Z}_q$. Then each of the singleton events $\{n_j\}$ forms a suitable partition, where each has an associated discrete $\Pi(\{n_j\})$. The integral of Definition 5.8 then becomes

$$\mu(d) = \max\left\{\min\{\mu(d, n_j), \Pi(\{n_j\}|d)\} \mid j \in \{1, \ldots, q\}\right\}. \tag{5.20}$$

Consider NPS $\simeq \mathbb{R}$. Then each of the individual Borel set $[a, b)$ (as discussed in Section 5.1.3) has an associated $\Pi([a, b))$, and a partition of the form $[n, n + dn)$ might be used. This can usually be characterized as a possibility density function $\pi(n)$. The integral of Definition 5.8 then becomes

$$\mu(d) = \sup\left\{\min\{\mu(d, n), \pi(n|d)\} \mid n \in \text{NPS}|d\right\}. \tag{5.21}$$

If all the points in the noise space are equally possible, ($\pi(n) = 1 \; \forall \; n$), then the development reduces to a simple max of $\mu(d, n)$ across the noise and design space, *i.e.*, this reduces to finding the max of $\mu(d)$ across the design space, as shown in Definition 4.4.

As an example, consider a one parameter design, with a μ preference function and a π possibility function as graphed in Figure 5-3. The maximum expected preference, *i.e.*, the maximum of the $E[\mu]$ function (using the possibilistic expectation of Equation 5.21), not the μ function maximum, is found. This set is a point d^* as shown on the design parameter axis.

5.3.2 Discussion

Recall that possibility represents uncertainty in the capacity of a design, which can reflect uncontrollable subjective uncertainty: the subjective evaluations of those other than the designer. This uncertainty, however, remains subjective uncertainty. This observation then poses the question whether the uncertainty should be resolved using a measure $\Pi_{\mathcal{P}}$, similar to the function $\mathcal{P}$ (Definition 4.2), rather than always using *min* (see Definition 5.8).

This question cannot be definitively answered. Subjective uncertainty remains an open research issue [23, 49, 57, 203]. Π is uncontrollable, however, and so the non-compensating *min* is practically a good choice. The development shown here corresponds with the historically developed possibility theory [50, 165, 196, 202], in the case of engineering design.

Another difficulty with possibility is the determination of the possibility specifications. These must be measured, as do all the uncertainty representations (μ and Pr, for example). If the possibilistic variables represent choices of others, these capacity limits are preferences of the other agents, and can be measured in a way similar to the way that the preferences of the designer were measured. Such possibility descriptions may also come from objective sources, such as material availability data, or adjustment limits within a design. In all cases, though, the designer adopts a belief in the stated degree of possibility Π when it is integrated into the determined overall preference. Thus, this again is a particular form of incorporating the judgments of others, and is not group decision making [7, 58].

5.4 Necessity

Noise modeled by necessity represents uncertainty within a formal model that the designer feels must be ensured, as defined in Chapter 3. The collection of all values of such parameters, as informally defined in Definition 3.8, form what is now termed a *necessity space*. The necessity space will be denoted NPS, meaning all uncertainties in the NPS are considered necessary, for this section.

Each necessary noise parameter has a domain percentage, as measured by the underlying uncertainty measure (either Pr or Π), that the designer feels must be ensured: if the noise parameter takes on a value within the domain percentage, the design will still function properly. Thus, there is a requirement of determining this domain (NPS) percentage required to be satisfied.

Definition 5.9 *A subset* $\mathcal{N}_\alpha \in \mathcal{B}$*, called the* necessary set, *is defined as the subset of the NPS that is desired to be satisfied.*

A problem thus becomes defining the set $\mathcal{N}_\alpha$. For the time being, this will be deferred, and it is assumed a subset $\mathcal{N}_\alpha \in \mathcal{B}$ has been identified.

Given the necessary set, any subset in $\mathcal{B}$ can be identified as necessary. If it lies within $\mathcal{N}_\alpha$, then the set is necessary, otherwise it is not. Thus, a necessity measure can be constructed given any necessary subset $\mathcal{N}_\alpha$ of the NPS. The measure can be defined by

$$\mathfrak{N}_\alpha(N) = \begin{cases} 0 & N \subseteq \mathcal{N}_\alpha \\ 1 & N \not\subseteq \mathcal{N}_\alpha. \end{cases} \tag{5.22}$$

Thus, only if all points within a set N are necessary does the set N become necessary, here necessary meaning $\mathfrak{N}_\alpha = 0$. The necessity measure therefore reflects the concern a designer has over the set being measured. Thus a low (zero) necessity reflects that the designer is concerned about the set; a high (one) necessity reflects that the designer is not concerned about the set. Thus the measure reflects not the degree of necessity, but the degree of designer concern over the set. $\mathcal{N}_\alpha$, however, must still be defined somehow.

Definition 5.10 *Any measure g satisfying Equation 5.22 will be called a* necessity measure, *and is denoted by* $\mathfrak{N}_\alpha$.

The necessity measure is thus a $\{0, 1\}$ placed measure. The designer uncertainty will be incorporated into α, to determine the extent of $\mathcal{N}_\alpha$. Thus the uncertainty is incorporated into specification of $\mathcal{N}_\alpha$ (the necessary set), not in $\mathfrak{N}_\alpha$ (the measure of a set in $\mathcal{B}$).

A necessary measure is not a fuzzy measure, it fails to satisfy the conditions of Equation 5.16. It is what will be termed a *dual*-fuzzy measure[1]. That is, the set inclusion

[1]This term is in keeping with the *principle of duality* of set theory [65], observing the reversal of the set nesting.

property is reversed from a fuzzy measure g. $\mathfrak{N}_\alpha$ instead obeys:

$$\begin{aligned} &i) \quad \mathfrak{N}_\alpha(\mathrm{NPS}) = 0 \quad \mathfrak{N}_\alpha(\emptyset) = 1 \\ &ii) \quad N_i \subseteq N_j \Rightarrow \mathfrak{N}_\alpha(N_i) \geq \mathfrak{N}_\alpha(N_j) \quad N_i, N_j \in \mathcal{B}. \end{aligned} \tag{5.23}$$

Also, clearly $\mathfrak{N}_\alpha$ is not necessarily continuous.

$$iii) \quad \lim_{j\to\infty} \mathfrak{N}_\alpha(N_j) \neq \mathfrak{N}_\alpha(\lim_{j\to\infty} N_j) \quad \text{for nested sets } N_j \text{ in general.} \tag{5.24}$$

An uncertainty integral can be constructed with the necessity measure. The integral of performance becomes the worst case performance across the necessity space, as measured by the necessity measure.

Definition 5.11 *Given a necessity space* $(\mathrm{NPS}, \mathcal{B}, \mathfrak{N}_\alpha)$, *the preferential performance of a point* $d \in \mathrm{DPS}$ *is defined by*

$$\mu_\alpha(d) = \inf\{\max\{\mu(d,n), \mathfrak{N}_\alpha(N)\} \mid N \in \{N_j\} \; disjoint \in \mathcal{B}\}$$

where $\mu(d,n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$, $n \in N$.

It can be verified that this definition is indeed an uncertainty integral. Again, the limit as the partition becomes finer in $\mathcal{B}$ is used. Thus, $\mu_\alpha(d)$ is the necessary expectation of $\mu(d,n)$ across the necessary space with respect to the necessary measure $\mathfrak{N}_\alpha$.

Now an expected preference is determined for each design parameter $d \in \mathrm{DPS}$, given the uncontrollable NPS. A performance rating has been obtained. Thus, the performance improvement principle can be invoked, and the highest performing design parameters defined. Definition 5.4 implies the most preferred points $\{d^*\} \subseteq \mathrm{DPS}$ are defined by

$$d^* : \mu_\alpha(d^*) = \sup\{\inf\{\max\{\mu(d,n), \mathfrak{N}_\alpha(N)\} \mid N \in \{N_j\} \text{ disjoint } \in \mathcal{B}\} \mid d \in DPS\} \tag{5.25}$$

where

$$\mu(d,n) = \mathcal{P}(\mu_1, \ldots, \mu_N)$$

is the overall preference for a point $d \in \mathrm{DPS}$, given the noise fixed at $n \in N$. $\mu(d^*)$ is the overall expected preference for the most preferred solution.

The problem is thus well formed, provided the set $\mathcal{N}_\alpha$ can be identified. How much of the NPS should be ensured? For an NPS $\simeq \mathbb{R}^q$ independent, this has typically been done with *confidence intervals* [168]. This formalism, however, assumes that the underlying uncertainty measure (Pr or Π) is constructed from a density function (*pdf* or π), which requires an ordering on the NPS. In any case, the percentage of the uncertainty space to be satisfied is identified as the *confidence factor*, and will be denoted α.

5.4.1 Probabilistic Necessary Parameters

When the underlying uncertainty is probabilistic, the density function that can be used to partition an ordered NPS is the probability density function, $pdf : \text{NPS} \rightarrow \mathbb{R}^+ \cup \{0\}$. The *pdf* can be used to define the necessary subset of the NPS, for any confidence factor α:

$$\mathcal{N}_\alpha = \{n \in \text{NPS} \mid pdf(n) \geq \Theta\} \tag{5.26}$$

where

$$\Theta = \inf\left\{\theta \;\middle|\; pdf(n) \geq \theta \text{ and } \int_{NPS} \chi_{\{pdf(n)\geq\theta\}}(n)\, dPr \leq \alpha\right\}. \tag{5.27}$$

Thus, Θ is the lowest level *pdf* value with α equal to the Pr of all n whose $pdf \geq \Theta$. This forms the set of elements n whose total $Pr = \alpha$, and whose n all have $pdf \geq \Theta$.

For example, in the case of NPS $\simeq \mathbb{R}$, and *pdf* as the normal distribution, $\mathcal{N}_\alpha$ becomes a class interval:

$$\mathcal{N}_\alpha = [E - r, E + r]$$

where E is the expected value, and r is a radial distance from the expected value such that $Pr([E - r, E + r]) = \alpha$.

In the discrete case, define

$$pdf(n_j) = Pr(\{n_j\}).$$

Then

$$\mathcal{N}_\alpha = \{n_j \mid pdf(n_i) \geq 1 - \alpha\}$$

can be used; *i.e.*, $\Theta = 1 - \alpha$.

The most preferred design parameter set and the maximal preference in light of probabilistic necessary parameters n then becomes:

$$d^* : \mu_\alpha(d^*) = \sup \{\inf \{\mu(d,n) \mid n \in \mathcal{N}_\alpha\} \mid d \in DPS\}. \tag{5.28}$$

To see this definition's meaning, consider a one variable design with a probabilistic necessary distribution, as graphed in Figure 5-4. Each value along the design parameter axis is uncertain because of the normal probabilistic variation as shown. Therefore the preference μ must be reduced at each point d to the lowest preference on the interval spanned by $d + \delta d$, where δd lies within the necessary range. For example, as graphed in Figure 5-4, at a design parameter value d_0, a 95% confidence interval ($\alpha = 0.95$) can produce variations in the range from d_- to d_+. The lowest preference μ in that range $[d_-, d_+]$ becomes the μ_α for d_0. This is repeated for all design parameter points to obtain the μ_α function. The maximum of this μ_α function is the solution: the most preferred design parameters subject to the necessary probabilistic distribution.

5.4.2 Possibilistic Necessary Parameter

When the underlying uncertainty is possibilistic, the density function which can be used to partition the NPS is the possibility density function π : NPS $\rightarrow [0,1]$. π can be used to define the necessary subset of the NPS, for any confidence factor α:

$$\mathcal{N}_\alpha = \{n \in \text{NPS} \mid \pi(n) \geq 1 - \alpha\}. \tag{5.29}$$

The most preferred design parameter set and the maximal preference in light of possibilistic necessary parameters n then becomes:

$$d^* : \mu_\alpha(d^*) = \sup \{\inf \{\mu(d,n) \mid \pi(n|d) \geq 1 - \alpha, n \in \text{NPS}\} \mid d \in \text{DPS}\}. \tag{5.30}$$

This definition is illustrated in Figure 5-5. Here π denotes the necessary region, and n is considered as the design parameter ($n = d$). Therefore the solution is the entire support of the π distribution, and the object is to rank the degree of preference for the range. Therefore

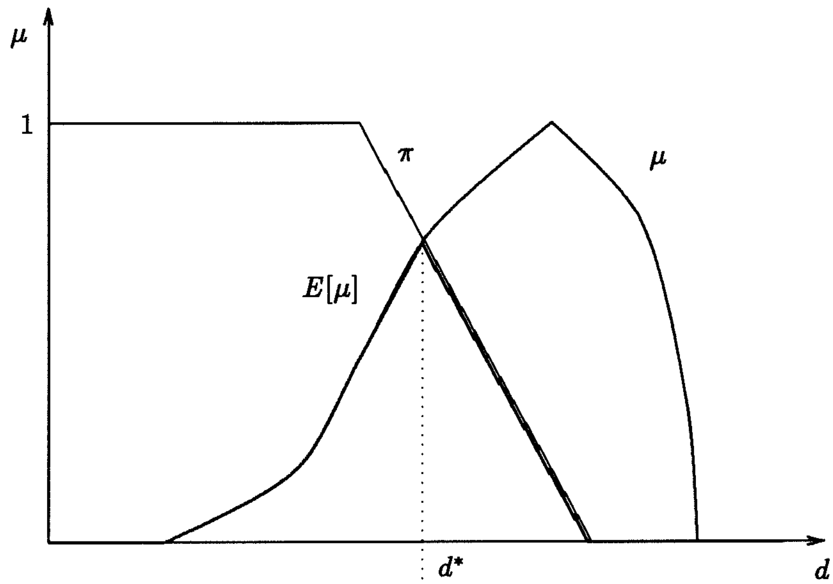


Figure 5-3: Parameter resolution with possibilistic uncertainty.

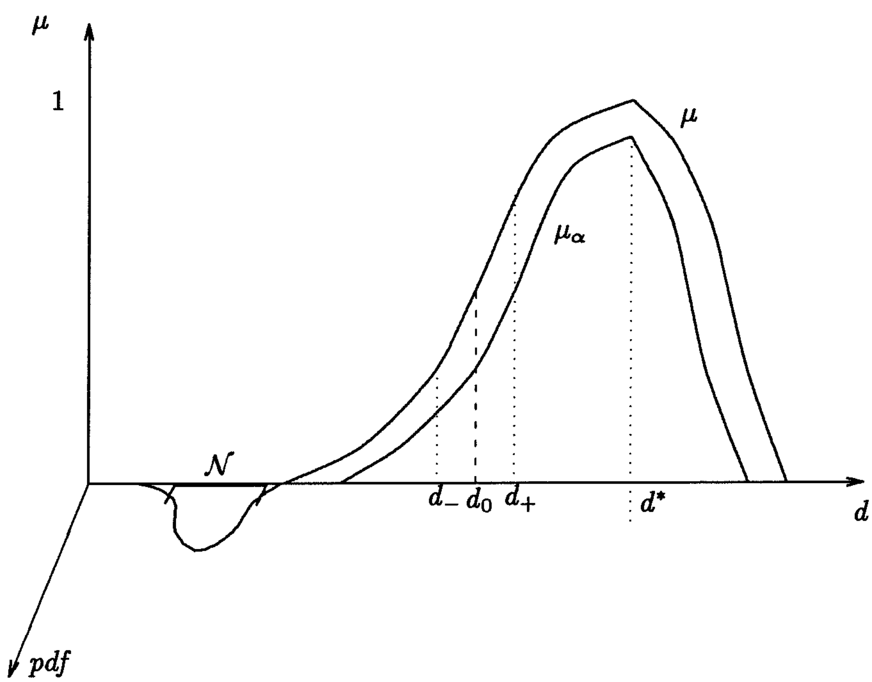


Figure 5-4: Probabilistic necessary parameter preference resolution.

the preference μ_α must be the lowest preference within the support of π. For example, as graphed in Figure 5-4, at a degree of necessity α, the domain of necessity is from d_- to d_+. The lowest preference μ in that range $[d_-, d_+]$ becomes the μ_α for the necessary range.

5.4.3 Discussion

Care should be taken with necessary spaces. If the development outlined in this section is taken, it must be understood that one is *not* designing to the average. If the underlying uncertainty is probabilistic and the standard expected value E is determined, E may not lie in $\mathcal{N}_\alpha$. The NPS must be examined to ensure this is appropriate.

For example, consider again an NPS $\simeq \mathbb{R}$, with split probability density, which results in performance as graphed in Figure 5-6. In this case, the necessary formulation is clear. The probable values are ensured.

Consider a density skewed from the average, however, that results in performance as shown in Figure 5-7. With the necessary formulation, the events with most chance are ensured first. Thus, with a sufficiently small confidence level α, the average may not be ensured. For example, $\mathcal{N}_{0.05}$ does not contain the average E.

Also notice a difference between the definitions of $\mathcal{N}_\alpha$ for the different forms of probability and possibility. Probability divides the NPS by percentage of area of the *pdf*. Possibility divides the NPS by percentage of height of the π. This reflects the difference in the uncertainty: probability is a cardinal rank, reflecting uncertainty of occurrence relative to the other values. Both an ordering of the elements and an intensity is required to form a probability measure. How much more probable one value is with respect to another must be known to form a probability measure. Possibility, on the other hand, is an ordinal rank, reflecting uncertainty in a parameter assuming a value. Only an ordering of the elements is required.

5.5 Hybrid Uncertainty

For problems with multiple uncertainty forms, Equations 5.15 through 5.30 must be combined. Such a combination is possible, but requires making explicit the *precedence relation*

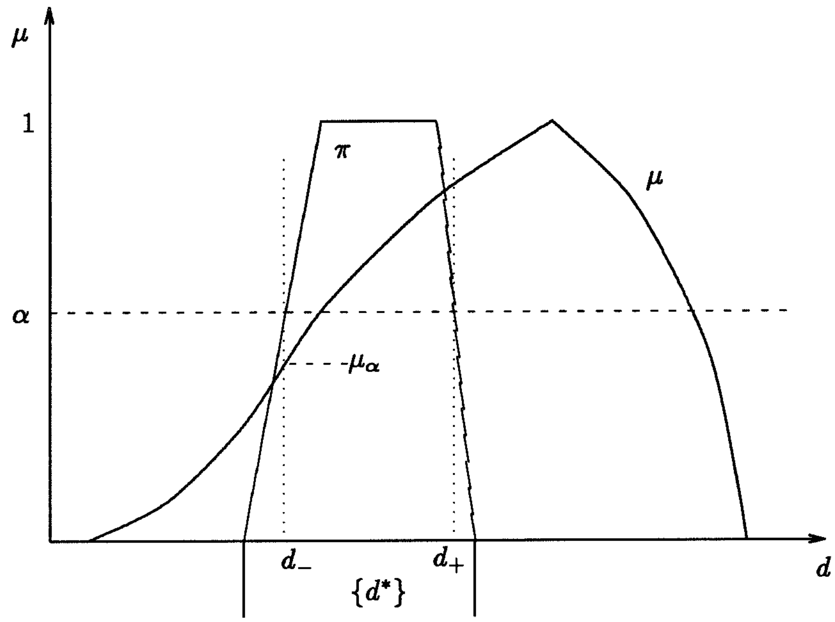


Figure 5-5: Possibilistic necessary parameter preference resolution.

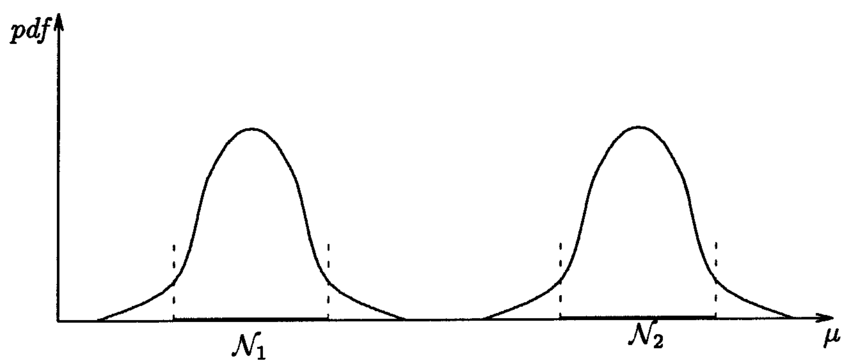


Figure 5-6: Double density example under necessity.

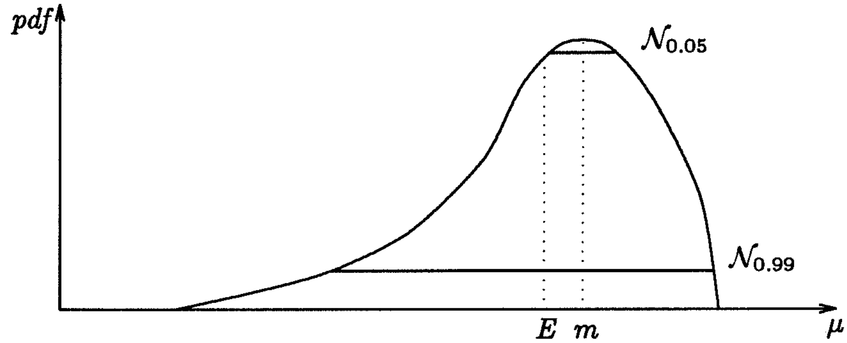


Figure 5-7: Skewed density example.

among the parameters.

As an introduction, consider again Example 1, the accelerometer example, introduced in Section 3.1 and shown in Figure 3-1, where the accelerometer indicates accelerations above a threshold with a switch closure. Recall, under specified accelerations, the accelerometer mass must contact a switch within specified time durations. The design is complicated by inaccuracies in the value for k, the actual spring constant. This uncertainty occurs randomly. Hence due to the manufacturing process, it is difficult to set precise performance parameter values of actuation times (time for the mass to close the actuation switch), and spring pre-load. Recall the design has a method, however, to overcome these manufacturing errors in the spring. Specifically, during manufacturing, the backstop of the mass can be adjusted to compensate for variations in k. This backstop positioning distance is a tuning parameter of the design, a possibilistic uncertainty. During manufacture, the spring constant of every accelerometer is measured, and the backstop of every accelerometer is positioned accordingly to meet the specified actuation times.

Such relations between design, confounding, and tuning parameters are readily modeled in the framework developed here. A tuning parameter's range of possibility forms a possibilistic uncertainty. There is a problem, however, in determining the order in which to combine the operations of the previous development.

In the accelerometer design, there were manufacturing errors in the spring constant k that could be overcome by the backstop position tuning. If the spring material is sensitive to the operating temperature, however, k will also vary with the operating temperature. Temperature is another noise parameter. Yet the tuning parameter (backstop positioning) cannot be adjusted to overcome this noise, the backstop is already positioned. All that can be done is to adjust the tuning parameter to maximize the performance across the *expected* temperatures, paralleling what a designer does when selecting design parameters. Hence relative to this component of the noise (temperature), the tuning adjustment is not a tuning parameter, but rather a "design parameter" of the manufacturing engineer. It's value is set (by the manufacturing engineer) to maximize the expected performance over the temperature noise. So the precedence relation in this design is: the manufacturing noise occurs, then the tuning parameter adjustment occurs, and then the temperature noise occurs.

Definition 5.12 *The temporal order in which specific values are determined for the parameters in a design, be they design parameters, noise parameters, or tuning parameters, is the* precedence relation *among the variables.*

Typically, the precedence relation is: design parameters are determined first, noise parameters occur subsequently in the attempt to achieve the design parameter targets, and finally tuning parameters occur to bring the design parameters back on their target values, despite the noise. The accelerometer example will be returned to later.

With the precedence relation established, one can combine Equations 5.15 through 5.30. One must take care to combine the equations in their proper order: design parameters on the outside, and tuning parameters on the inside (relative to the confounding noise parameters).

It is not always true that possibilistic uncertainty has its evaluation inside the probabilistic integral. If there had been possibilistic uncertainty associated with the design parameters d to limit the designer's choice, then this combination occurs outside the manufacturing error integral. Similarly, if the designer had particular preferences for tuning parameter values, then this combination occurs inside the integral. The precedence relation among the variables is determined by the temporal order of parameter specification.

For a general design problem, the evaluation order of the maximizations, minimizations, and integrals will depend on the precedence relation among the variables. This is not an artifact of the imprecision formulation. The same problem occurs with any other formulation (such as probabilistic optimization or Taguchi's method, extended to include tuning parameters [113, 119]). Imprecision simply sets the metric across the space to be μ rather than, for example, a single performance parameter expression. Using the additional restrictions of Section 4.7 on the overall preference metric to reduce to these other methods (optimization and Taguchi's method) provides the formalization of these methods with hybrid uncertainty; for example, for formalizing design problems with probabilistic noise and tuning parameters.

5.6 Examples: Designs with Uncontrollable Parameters

Now two examples are presented to demonstrate the definitions developed. The first example considers Example 3 of the thesis, the design of a pressurized air tank. The example is now expanded to include noise, including probabilistic, possibilistic, and necessary forms. The second example considers Example 1 of the thesis, the design of an accelerometer actuation switch. The example is now expanded to include noise, both probabilistic and possibilistic forms, and in particular will be an example of the use of tuning parameters.

5.6.1 Example 3: Parametric Design with Noise

The example presented herein considers Example 3, the design of a pressurized air tank. The example is now expanded, in that the previous formalization considered the noise parameters as constants. Now they are allowed to vary.

Recall the design problem is to determine length and radius values of air tanks with two different choices of head configuration: flat or hemispherical. See Figure 4-4.

This example's design space is formed by the 2 design parameters l and r, and so the DPS $\simeq \mathbb{R}^2$. The PPS is formed by the 4 performance parameters, v, m, L_0, and R_0 and so the PPS $\simeq \mathbb{R}^4$. The preferences for values of these design parameters and the four performance parameters are graphed in Figures 4-5 through 4-10.

The problem now, however, is confounded by noises. Manufacturing errors on l and r exist, for example, that limit the accuracy with which one can specify their values. This error is formalized as Gaussian. In this example, error is also introduced by the supplied material variability. This error is manifested in the allowable stress S, which is assumed to vary as a beta distribution. The effects of these errors are desired to be minimized.

Finally, there is error introduced in the variability of the welds made. This error is manifested in the joint efficiency E, which varies as a beta distribution. The effects of these errors must be protected against, since failure in a weld represents a safety concern. Therefore, this error is modeled as a necessary probabilistic uncertainty.

The other unknown in the problem is the applied pressure P, which can vary with use. This is represented as a range of possibilistic necessity from -15 and 120 psi, since this design must satisfy all these pressures.

These distributions are graphed in Figure 5-8. The necessary parameters used a value of $\alpha = 0.9$, or the variables were satisfied 90 percent of the time. These delimiters are also shown in the figure. The NPS, formed by δl, δr, S, E, and P, is thus a subset of $\mathbb{R}^5$.

The problem, then, is to find the values for l and r that maximize overall preference. The method of evaluation remains to be defined: a strategy must be set. In this particular example the non-compensating design strategy will be used. This means that among the multiple goals of the design, the worst performing goal will be improved, if any improvement can be made at all, by changing values of the design parameters l and r.

Applying Definition 5.6 and Definition 5.11, then for a non-compensating design strategy the problem to be solved is to find l^*, r^*, where

$$\begin{aligned}\mu(l^*, r^*) \quad = \quad & \sup \{\inf \{\textstyle\int_S \int_{\delta r} \int_{\delta l} \\ & \min\{\mu_l, \mu_r, \mu_v, \mu_m, \mu_{L_0}, \mu_{R_0}\} \times pdf(S)\, pdf(\delta r)\, pdf(\delta l) \times dS\; d(\delta l)\; d(\delta r) \\ & \mid P, E \in \mathcal{N}_{0.9}\} \mid (l, r) \in \mathbb{R}^2\}\end{aligned} \tag{5.31}$$

This will find the l^* and r^* that maximizes the preference of the design aspect with lowest preference, yet considers the confounding probabilistic noise effects, and satisfies the necessary parameters 90 percent of the time.

The design space is graphed in Figure 5-9 for the flat and hemispherical head design. The peak preference point represents the design parameter values to use: those with maximum expected preference, given the designer specified preferences, necessary distributions, and the noise distributions. The results are different from the case when no noise or necessity

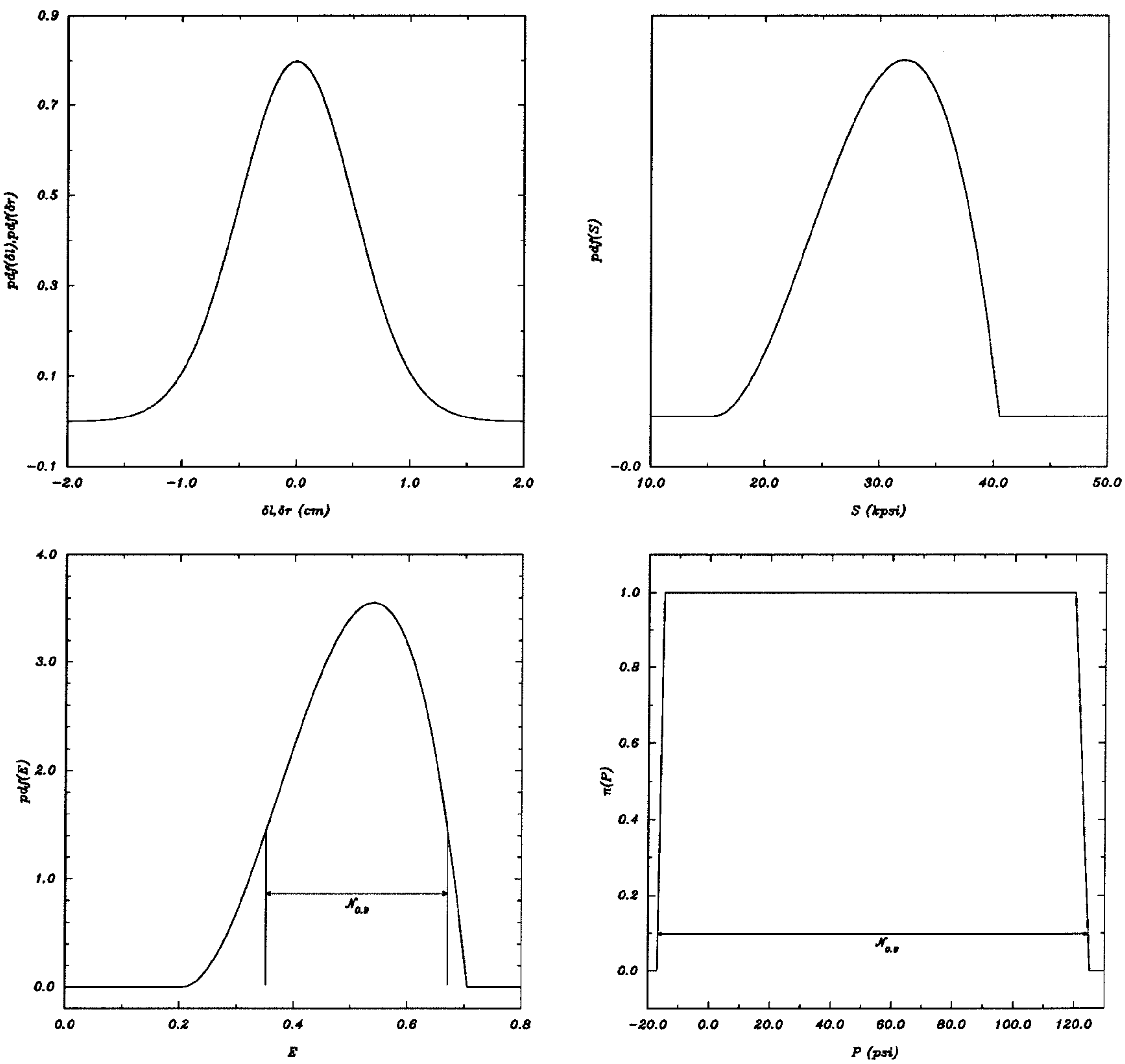
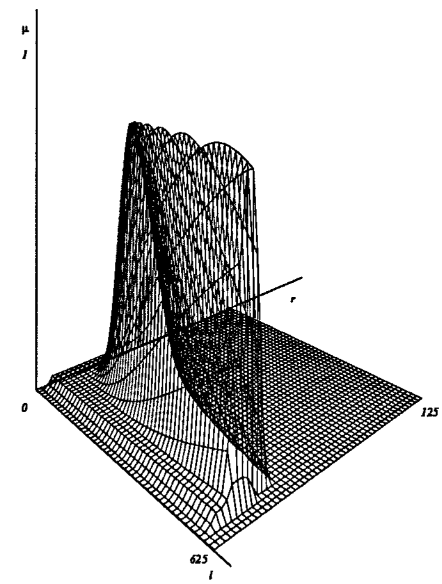
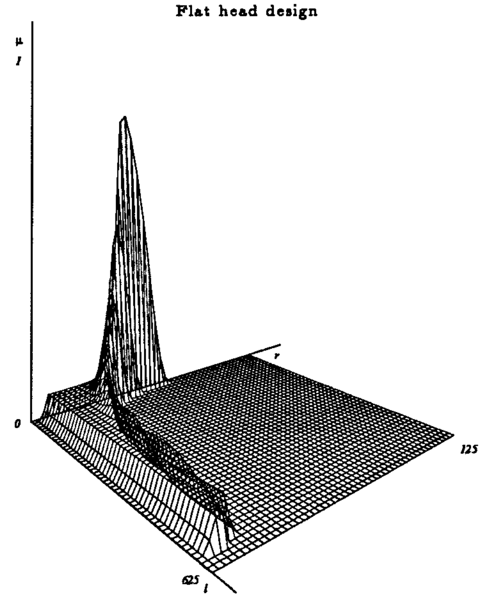


Figure 5-8: Example 3: noise distributions.



Flat head design



Hemispherical head design

Figure 5-9: Example 3: expected preference across the design space (r,l).

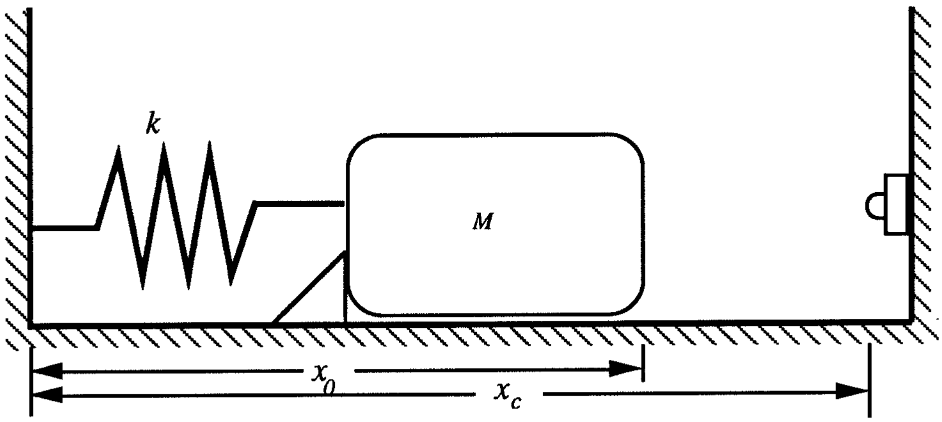


Figure 5-10: Example 1: accelerometer model.

was considered. When only preference information is considered (the expected values are considered for the noise parameters), the resulting l and r values are chosen directly on the imprecise constraint boundaries, as shown in Section 4.6.2. The consideration of noise backs the chosen parameter values off the imprecise preference curves to more robust values.

5.6.2 Example 1: Tuning Parameter Design

The example just presented demonstrated a design problem with the noise forms of probability, possibility, and necessity. Tuning parameters, however, were not exhibited by the design. Example 1, the accelerometer switch example, is now reconsidered, and formally developed.

In the accelerometer design (shown in Figure 3-1) there is a mass M attached to a spring k attached to the ground. The ground is accelerated. With sufficient acceleration, the mass must displace a specified distance to make contact with a switch. There is also a backstop placed against the mass, that the spring k pulls against with pre-load P under no acceleration. Refer to Figure 5-10.

There are two goals in this design: to maintain a specified preload P, and to close the switch in time τ under a specified acceleration. The parameter τ reflects the desired actuation time, and the pre-load P reflects the desired insensitivity to weak accelerations.

As a part of these goals, the designer needs to determine whether the design can be made sufficiently tolerant to variational noise to satisfy the customer.

There are two design parameters, mass M and spring constant K. There is, however, uncertainty in the manufacture of the spring: a random variation on K, denoted δk. Finally, to assist in maintaining the targets on the goals, the manufacturing procedure can position the backstop based on measurements made of the total spring constant ($k = K + \delta k$) of each accelerometer. This backstop distance is denoted x_0, and is a tuning parameter. The switch distance is denoted x_c. The position of the mass at any given time is denoted x. The mass is to make contact with the switch when subjected to acceleration a.

To determine the time to actuate the switch, the differential equation of motion of the mass must be solved. It is:

$$M(\frac{d^2x}{dt^2} + a) \times H(x - x_0) + kx = P \times H(x_0 - x) \tag{5.32}$$

where H is a step function, $x(0) = x_0$, and $\dot{x}(0) = 0$. This can be solved for the time to actuation:

$$\tau = \sqrt{\frac{M}{k}} \times \arccos\left(\frac{Ma - (x_c - x_0)k}{Ma}\right) \tag{5.33}$$

This solution assumes, of course, a is sufficiently large to move the mass (i.e., the *arccos* is defined). The other goal is the pre-load P, whose equation is also determined from the above differential equation:

$$P = kx_0 \tag{5.34}$$

Maintaining a specific pre-load helps eliminate spurious switch closures.

The problem is now formalized: the DPS is formed by the M and K variables, and so the DPS $\simeq \mathbb{R}^2$. The NPS is spanned by the variational noise on K and the tuning parameter x_0, and so the NPS $\simeq \mathbb{R}^2$. The performance parameter space is spanned by two goals, P and τ, and so the PPS $\simeq \mathbb{R}^2$.

Having formulated the problem, it can now be solved by formal methods. Using the methodology of this thesis, the design parameters and performance parameters will have associated preferences, as graphed in Figures 5-11.

There is also a random variation δk on K. The distribution of δk is graphed in Fig-

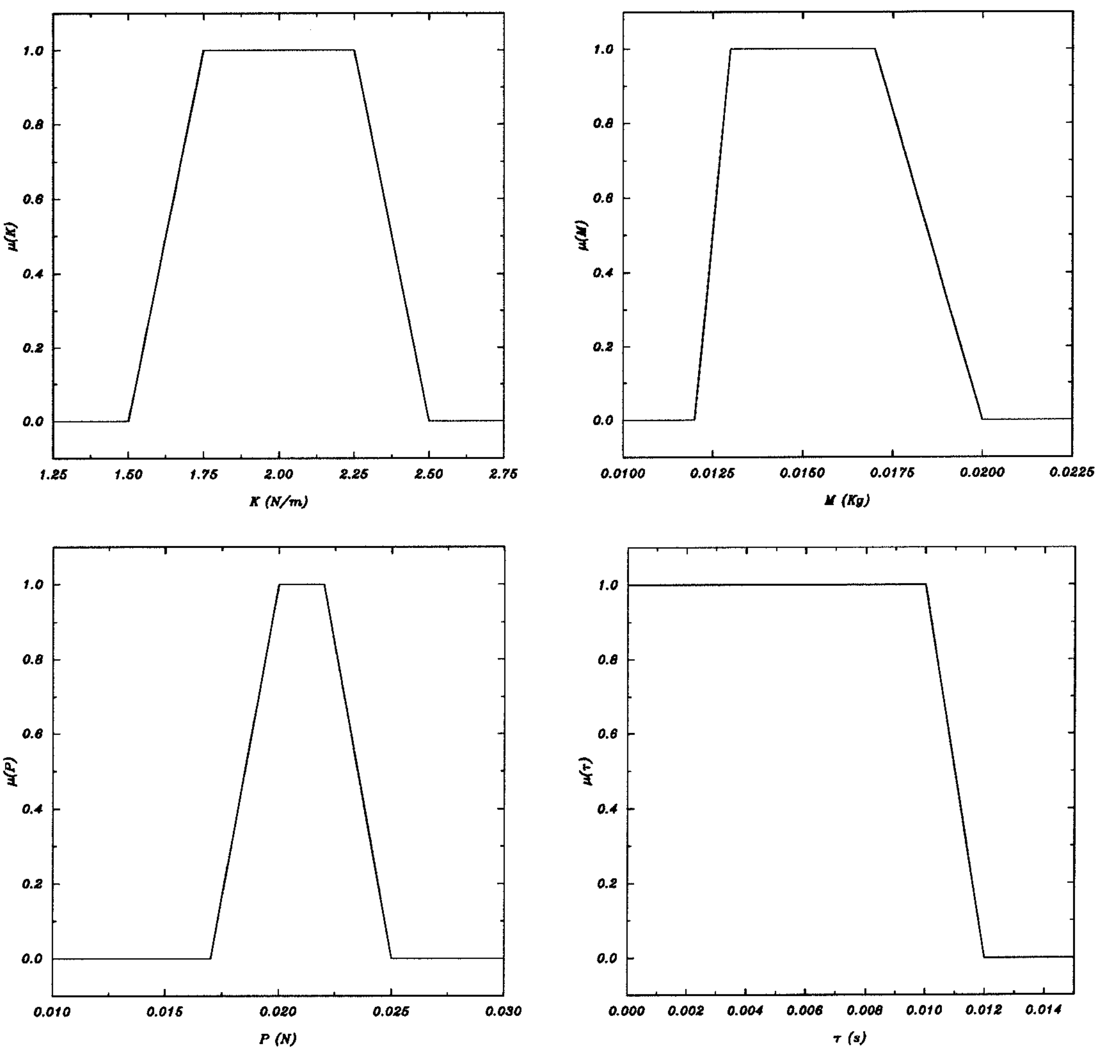


Figure 5-11: Example 1: M, K, P, and τ preferences.

ure 5-12. Also, there is an ability to tune the design with x_0. The manufacturing engineer will select the value within the model's range, whose bounds are crisp. There are definite physical limits on x_0. Therefore, the possibility distribution of the tuning parameter is a step function, as graphed in Figure 5-13.

The determination of these distributions now allows the design to be solved with the methods of this chapter. Again, as an illustration of one design approach, a non-compensating trade-off strategy will be adopted (Equation 4.5). Applying Equations 5.15 and 5.21 in conjunction with the precedence relation, the problem becomes to solve for M^*, K^* such that

$$\mu(M^*, K^*) = \begin{array}{l} \sup \{ \int_{\delta k} \sup \{ \min\{ \min\{ \mu(k), \mu(M), \mu(\tau), \mu(P)\}, \pi(x_0)\} \mid x_0 \in \mathbb{R}\} \\ \times pdf(\delta k)\; d(\delta k) \mid (M, K) \in \mathbb{R}^2 \} \end{array} \tag{5.35}$$

The applied acceleration is assumed to be $a = 20$ g's, constant. The preference surface is graphed in Figure 5-14, the supremum of which is the most preferred point. This surface reflects the designer's expected preferences, given the manufacturing errors and the ability to overcome them to within the range of the tuning parameter.

5.7 Related Work

Probabilistic noise has been under development for many years [28, 65, 79, 150], and the development presented here is in general agreement with this work. The preferential metric developed in Section 5.2 is the expected value of the preference across the noise, expressed by a Lebesgue integral. Halmos [64] discusses measure theory, or Royden [139] discusses more generally mathematical analysis.

The development of possibility has much to owe to past work on possibility measures in the fuzzy sets community. Zadeh [202] originally conceived of the theory of possibility in decision making. Sugeno developed the theory of a fuzzy measure and a fuzzy integral [164], in the form of the specific measure expressed in Equation 5.18. Also, the integral across a fuzzy measure space was defined, similar to Definition 5.8. Yager presents the differences between probability and possibility as cardinal and ordinal numbers, within the application

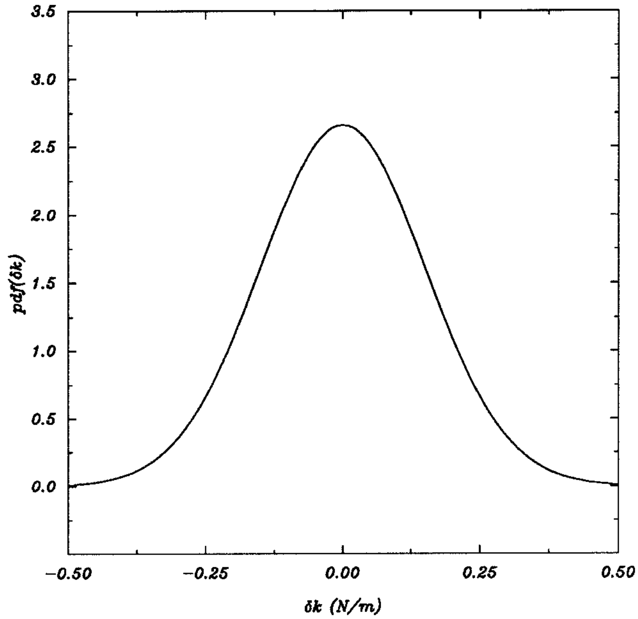


Figure 5-12: Example 1: random noise δk distribution.

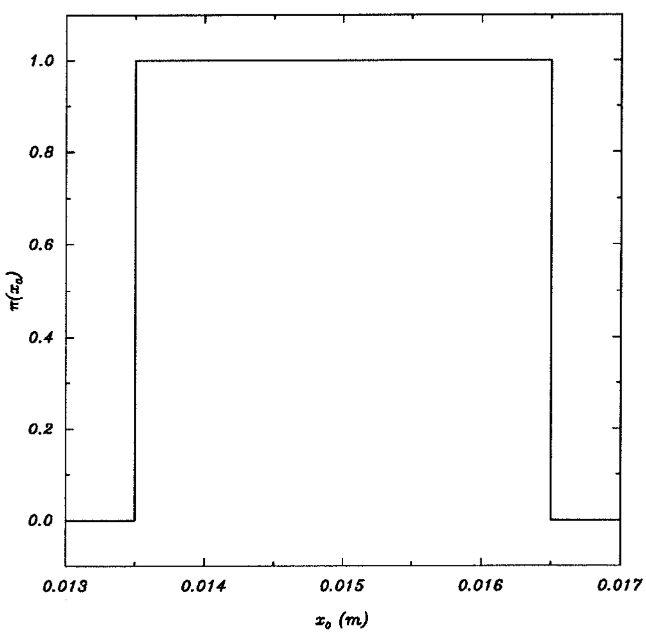


Figure 5-13: Example 1: tuning parameter x_0 possibility distribution.

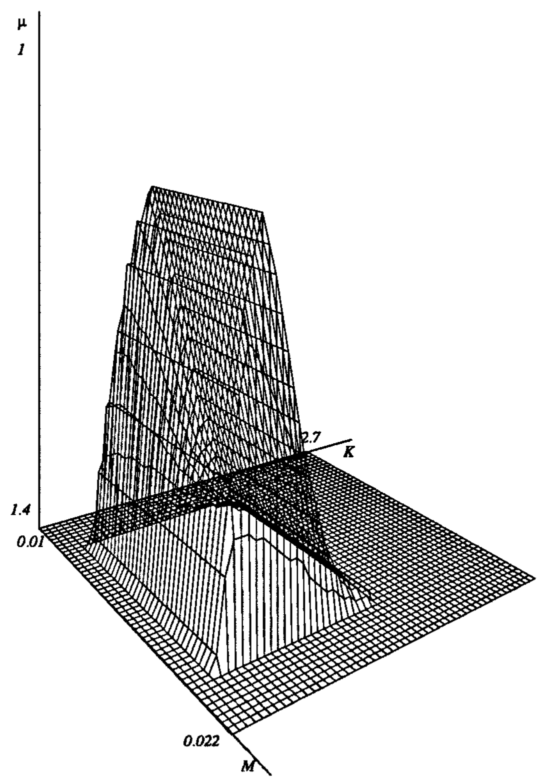


Figure 5-14: Example 1: expected preference across the design space (M, K).

of possibilistic decision making [196].

These works were considered by Dubois and Prade [43, 50] to form a more general theory of possibility and fuzzy measure, combining Sugeno's work and triangular norms, as originally presented by Menger [94]. The probabilistic and possibilistic integrals were also developed by Dubois and Prade. The more general uncertainty integral framework, presented here, generalizes this work.

Though these formulations have been previously determined, nowhere have the ramifications of convolving the separate theories been methodically studied, as is done here. Kaufmann and Gupta [83], however, do introduce "hybrid numbers," which are parameters with both a fuzzy and a probabilistic component, on which computations are performed. Wood [187] and Wood and Antonsson [189] extend this to include possibility as well. This development works well when the problem need not consider any precedence relations among the uncertainties, so the separate uncertainty forms can be calculated in parallel. This is not true in general.

Dubois and Prade [43, 50] present the same possibility measure as presented here. They present a different necessity measure, however, than that developed here. Dubois and Prade consider the informal meaning of an event being "perhaps not possible." They define this as necessity. This is different from necessity as defined here (Definition 3.8), which reflects the degree that a parameter must be ensured. Thus, their meaning of necessity and the meaning of necessity as developed here are entirely different.

Given that what is desired to be represented is different, it is no surprise that the formal mathematics developed are different. Dubois and Prade develop a formal measure *Ne* defined to be $Ne(N) = 1 - \Pi(\mathrm{NPS}/N)$. Such a measure can be used, but it provides the same information as the possibility measure, only in a different formalization.

Given this development, Dubois and Prade also discuss expected values of fuzzy quantities in the possibilistic, probabilistic, and their necessity environments [49, 50], relying on earlier integration work by Dempster [31]. Such work is therefore closely related to the developments presented here. The full ramifications of the development, though, is lost in their presentation. There is no consideration of hybrid uncertainty, as in Section 5.5. No previous development could be found regarding tuning parameters, for example, though they are obviously common to the practicing engineer.

The work in this section presents a method to evaluate an overall preference metric, despite the effects of noise. Another formalism for the same informal problem is utility theory. When the NPS is a probability space and when the DPS $\simeq X$ (*i.e.*, preferences can only be stated by considering all the goals simultaneously), then the development presented here reduces to utility theory.

When the formal problem is restricted to probabilistic spaces, others have claimed that maximizing the expected value is not the only formalism to adopt. Another method might be to instead rate the design by the variance across the NPS. In Taguchi's method [26, 73, 81, 138, 157, 169, 170, 177], for example, the "quality" of a design parameter set $\vec{d}$ is defined by the expected variation of a single performance parameter f from a target value τ. The difference between the developments is simply a difference in terminology. The performance parameter could as easily be defined as variation from target ($p(\vec{d}) = |f(\vec{d}) - \tau|$), and then Taguchi's method can be seen as rating a design by the average, as is done here. Therefore, Definition 5.6 and Taguchi's method are similar in the quality concept: Taguchi's method incorporates an experimental approximation to the integral in Definition 5.6 across probabilistic noise. The difference is that Taguchi's method is finding the mean of a single performance parameter, not preference over many design parameters and performance parameters. This is fully developed in [113], where Taguchi's method is extended to include tuning parameters. Methods for including tuning parameters in existing design techniques is further explored in [119].

The relationship between the development presented here and experimental design is demonstrated by observing that the integral of Definition 5.6 can be approximated by experimental points in the noise space; *i.e.*, experimental design techniques can be used. These points can be chosen using a factorial method. See [12, 20, 80, 98] for a discussion of factorial methods and methods for determining experimental points. Alternatively, Monte-Carlo simulation [66] could be used for more accuracy.

In the situation of the DPS $\simeq \mathbb{R}^n$, probabilistic optimization methods can be used to evaluate the solution, as presented by Siddall [156, Chapter 13]. Again, it must be made clear that the development discussed here is as difficult to evaluate as the non-linear probabilistic programming formulations, in general. Definition 5.4 defines a constrained optimization problem: maximize μ subject to $\vec{d}$ being within the support of μ, with noise as characterized by the NPS. Ensuring this constraint is non-trivial, and will, except in the easiest of problems, require the nonlinear probabilistic programming formulations such as a boundary method [156, Chapter 13], to gradually incorporate preferences that initially are zero.

There has been other research work in the mechanical design domain with possibility and necessity. Ward [182] and Ward and Seering [184] have developed a "Labeled Interval Calculus" that consists of interval mathematics with associated "only" or "every" labels. The formal relation between Ward's work and the work in this thesis remains unexplored, but informally the intentions of the formalisms are similar: to formally specify the different intentions of a designer.

When the NPS has been formalized, a decision must be made whether the noise should be rated by the expected value, or by the worst case scenario (*i.e.*, whether to model the uncertainty as necessary). Others have viewed the concept of necessary parameters with disfavor. Consider the case of probability, which has received the most attention. Siddall [156, Appendix C] or Savage [150] reject using confidence intervals. Siddall argues that one has difficulty determining the density function *pdf* and the confidence level α. Clearly the first concern is always present, and applies to design problems as modeled in Section 5.2 as well, which Siddall advocates. The concern over the determination of the confidence levels, however, must be addressed. This is a subjective determination of the designer. A solution to the specification difficulty is that the designer cannot specify such values *a priori*, but must informally iterate with the formal model to settle on values, trading off the confidence level α with the overall degree of preference μ.

Savage [150] argues against the use of confidence information in that, if the formalization is accurate, knowledge about any variation is superfluous, since he feels one is still obliged to choose the expected value. This is not true for engineering design, where one must decide whether to maximize the average, or maximize the worst case. Also, this obligation is also true only if one has absolute belief in the initial formalization, which is almost never true. Savage also argues against confidence intervals in the statistical domain of tests [150]. Savage argues against the idea of having a specified probability α of failing the test hypothesis. That is, he argues against it being possible to design a test such that if a test hypothesis holds, then the probability that this is in fact correct for each $n \in \mathcal{N}_\alpha$ is α. This is different from the idea of necessity, which calculates a design such that the probability of the design being ensured is α. The probability of any $n \in$ NPS as well being in $\mathcal{N}_\alpha$ is α. Thus, the two ideas are completely different.

Though theoretical concerns do exist with necessary parameters, the fact remains that confidence intervals are used in practice [1, 68, 91], and modeling to consider the worst case noise as well [40]. Dismissing such modeling due to present theoretical limitations is misdirected, and instead should call for development and further research.

5.8 Conclusions

A NPS formally models what the designer does not have control over, yet must account for when evaluating performance. As such, the designer must consider all values in the NPS, and rate their effects. Such a process is formally characterized by integration across the NPS.

A feature of the NPS as compared to the DPS is the structure required. The NPS must form an uncertainty measure space. The DPS, on the other hand, required set structure with preferential orderings. The measure structure of the NPS is reflecting that if a designer cannot control the parameters, the designer must at least be able to characterize them, in order to handle them within the design. This chapter demonstrates a level of characterization required. The parameters must be understood to the point where they can be characterized with an uncertainty measure space. The difference is actually more apparent than real. To actually construct preferential specifications on the DPS, typical methods have topological structure, as presented in Appendix A.

The formal mathematics developed for each parameter form are not only founded in informal justifications, there are also formal foundations. The integration operations developed have algebraic structure. That is, the measure and its associated integration operator are derived from t-norm ($\perp$) and t-conorm ($\star$) pairs. They have the properties of commutativity and associativity, and are related through negation:

$$g(N_i) \perp g(N_j) = \neg(\neg g(N_i) \star \neg g(N_j))$$
$$g(N_i) \star g(N_j) = \neg(\neg g(N_i) \perp \neg g(N_j))$$

The $*$ (multiplication) of the probability measure is a t-norm, and the $+$ (addition) of the Lebesgue integral is its associated t-conorm. Likewise, the *min* of the possibility measure and the *max* of the possibility integral are an associated t-norm/t-conorm pair. The *min* of the necessity integral and the *max* of the necessity measure is then dual to this pair. But in all the integration procedures, there is an underlying mathematical structure. Thus, uncertainty integrals have a canonical form, as expressed in Definition 5.2.

Given a noise space, this chapter was devoted to methods by which a designer can define a solution. There are other operations that a designer may wish to do, however. For example, given a probabilistic NPS, the higher moments of the distributions provide sensitivity information, which is useful to a designer in assessing performance. Target levels for sensitivity may be determined, and thereby a new PPS formed. Analysis of variance (ANOVA) [12, 20, 98], for example, is an experimental method for determining such additional information.

The decision over the method of resolution, given an uncertainty distribution (as a necessary parameter or not), is a subjective evaluation which the designer must make informally. The concept is new, however, and little work has been done to assist the designer in such decisions. Sometimes the decision is made for the designer by customer requirements. For example, Motorola has a corporate drive to contain its manufacturing errors to within 6 standard deviations [1, 68, 91]. Here, the corporation has made the decision for the designer. It may not always be clear, however. System design techniques, as described by Wilson and Wilson [185], separate goals into those that must be satisfied and those that are desired to be satisfied. Those which must be satisfied could rightfully be modeled with necessary restrictions on noise. It is not clear for those goals which are desired to be satisfied. Additionally, examples such as the double density distribution (Figures 5-2 and 5-6) and the skewed distribution (Figure 5-7) make even this breakdown less justified.

Determining when a parameter is a tuning parameter, on the other hand, is more direct.

A tuning parameter is one whose value is set to counter-act any noise in the design. The tuning parameter's value is a possibilistic uncertainty from the design engineer's perspective. It has a range of possible values and the value that should be used cannot be set by the design engineer, since it will depend on the manufacturing errors. But the expected value of the probabilistic manufacturing error can be determined. Since the possibilistic tuning parameter's values depend on the probabilistic manufacturing error, then from the design engineer's viewpoint (pre-manufacturing) the tuning parameter expected value can also be found. That is, the possibilistic tuning parameter will adjust its value based on the probabilistic manufacturing error. Hence there will be, from the design engineer's viewpoint, a probabilistic distribution for the tuning parameter as well, even though it has no probability aspect associated with it at all. Taking a view of the tuning parameter before the noise occurs, it can be said to have a probability distribution. But, inherent in the parameter itself (*i.e.*, from the manufacturing engineer's perspective who must actually set the variable's value after the noise has occurred), the parameter has absolutely nothing to do with probability.

This chapter has presented different formal schemes to represent different uncertainty forms that can exist in a design. Given uncontrollable parameters in a design, the methods of this chapter can be used to identify the design that formally provides the designer with highest satisfaction, despite the confounding effects of noise.

Chapter 6

Preliminary Design

The previous two chapters have dealt exclusively with defining what should be determined in a formal engineering model. It has been stressed, but not developed, that any such determination requires iteration. An iterative loop is assumed whereby informal conjectures are formalized, formal calculations are made based on these informal conjectures, and then subsequent informal interpretation of the formal calculations occur. This usually induces more changes in the informal model, and the informal/formal iteration process cycles. Such an interactive design process will now be given formal foundation, as developed and presented in [107, 108, 118, 191, 192].

Section 6.1 informally discusses what is meant by preliminary design and partial specification, using the development of this thesis. Section 6.2 presents the formalization of partial specification. Section 6.2.2 then presents computational algorithms for the particular case of DPS $\simeq \mathbb{R}^n$ and PPS $\simeq \mathbb{R}^m$ independent, and examples shown. Section 6.2.3 presents further examples, but which exhibit the effects of singularities on the induced preference. Section 6.2.5 presents examples when the performance maps are expressed as differential equations. Section 6.3 leaves the demonstration of induced preference, and considers the consequences of the definitions. Conditions are shown under which the induced preference is invariant of the strategy used. When every goal has a preference specification, Section 6.4 demonstrates how to use induced preference to calculate the most preferred design parameters, as discussed in Chapter 4, thus tying the developments of the two chapters. Section 6.5 then introduces noise into the induced preference definition, thus tying this chapter with

6.1 Introduction

A hypothesized description of an iterative design process is that a designer may desire to partially specify a model. In the preliminary design phase, after a formal model (a DPS, PPS, and NPS) has been constructed, the model may be iterated over: a designer may specify particular values on *some* of the parameters, simply to observe the effects of the partial specification.

For example, a designer may wish to conjecture values for design components, and observe the restrictions on performance such specifications induce. In a design involving material, selecting particular ranges of values for the design geometry will restrict the range of possible stress that can occur. Performing this calculation (determining the restricted range of possible stress) will allow a designer to interpret the performances achievable if the design configurations sought by the designer are pursued.

On the other hand, a designer may wish to conjecture values of design performance, and observe the restrictions over the design options that can be used. Selecting particular ranges of values for stress will restrict the range of possible geometries that can be used. Performing this calculation (determining the restricted range of possible design configurations) will allow a designer to interpret what design configurations should be pursued if the performances desired are to be achieved. Of course, such configurations may be impossible. Thus, a designer can iterate among the specifications.

This chapter will now provide a formalization of the process of partial specification. This will be pursued by allowing preference specifications μ to be made on some of the parameters. Given such preference specifications, the restrictions that such μ impose will be propagated through the model. Thus, the development will involve formal computation and informal interpretation: the propagation of initial preference specifications will be formally computed, and the induced restrictions requires informal interpretation.

One can ask, however, if providing such a formalization is theoretically acceptable. The results obtained may be path dependent: specifying one aspect before another may lead to different results. Perhaps it is always better to instead simultaneously elicit preferences on *all* of the parameters at the start of a design process, and then combine them as discussed in Chapter 4. The problem with such an approach is that any determined preference is known to always be dependent on the context in which it was generated [13, 158, 178]. Thus, the results obtained may not be accurate. Those who espouse such methods realize the formalization may not be accurate (see French [58], for example), and so caveat their methods by insisting that they are simply guidance structures that allow insight into a decision making process, and can be modified. Thus, initial formalizations are always assumed to be conjectured, and iteration is implied.

It is believed that a way to overcome the path dependency problem is to pursue many paths. A design process should allow any specification to occur (in a context, as all specifications are), and then allow such specifications to subsequently be modified with new insight. Again, it must be emphasized that this is an informal/formal iteration.

Assumption 6.1 *A viable design activity is to* partially *specify a design, to observe the effects of the* a priori *partial specification, which can then guide a design process into subsequent re-specification and eventually to a full specification of the design.*

This assumption is based on observations of engineering design practice: preferences for parameter values are formed based on the design activity.

The design process characterization presented here will thus involve informal interaction with a formal model. It is not claimed that what will be developed is *the* design process, if there ever can be such a characterization. Rather, *a* particular characterization will be developed of *a* design process, which is claimed to have particular elegance, both theoretically and in utility. It is hypothesized that informal/formal modeling interactions *always* occur in design; it is a canonical activity. It is believed that any elicitation of this activity will have practical utility. This informal/formal iteration described will now be developed in the context of the previous chapters: using preference.

6.2 Induced Preference

A *semi-automated* approach is pursued to the design of complex mechanical entities as a process. It is proposed to aid the designer's ability to make decisions in the preliminary design stage with formal computation. To do so, it is assumed that formal computation can occur: the designer has formalized a design problem into a formal model, as discussed in Chapter 2. Thus, there is a DPS, PPS, and NPS. The NPS is assumed trivial for the time being.

Yet the formal model so determined is not completely described: the designer cannot determine which candidate offers the best alternative to pursue. The complete preference specification has not been made.

It is assumed, however, that the designer can make preference specifications on *some*, but perhaps not all, of the parameters. That is, individual preferences μ (as discussed in Chapter 4) have been determined for some of the parameters. Given the *a priori* conjectured preferences, a formal calculation which could be performed is to induce the specified parameter preferences through the performance maps to the unspecified parameters, and thereby observe *all* of the possible achievable performances on these unspecified parameters.

The description just provided will now be given formal structure. The total imprecise space is DPS $\times$ PPS. In general, each space itself is assumed to have a partial specification, thus each space is separable into specified aspect S, and an unspecified aspect U. That is, each space is assumed to be separable into a Cartesian product,

$$\text{DPS} = S_d \times U_d \tag{6.1}$$

and

$$\text{PPS} = S_p \times U_p \tag{6.2}$$

where preferences are specified on S_d and S_p, but U_d and U_p remained unspecified. Let μ_d denote the preference specification on S_d, and μ_p denote the preference specification on S_p. To induce the preferences onto the unspecified spaces, first these two preferences must be combined. Combining preferences involves a preference metric, as argued in Chapter 4. Thus, the issue of trade-off exists in the preliminary design phase as well.

Let $\mu(s_d, s_p) = \mathcal{P}(\mu_d, \mu_p)$, where $\mathcal{P} \in IMC([0,1]^2)$ (Definition 4.5). This result is the partial specification discussed in the introduction, here expressed with preference. The desire is to now induce this preference specification from the specified space S onto the unspecified space U.

To do so, recall the relation between the DPS and PPS as formalized in Chapter 2. There is a set map $f : \text{DPS} \rightarrow \text{PPS} : d \mapsto p$, which will now be expressed using the components ($d = (s_d, u_d)$, $p = (s_p, u_p)$):

$$\begin{aligned} f \quad : \quad & S_d \times U_d \rightarrow S_p \times U_p \\ & (s_d, u_d) \mapsto (s_p, u_p). \end{aligned} \tag{6.3}$$

Now define $\text{S} = S_d \times S_p$, and $\text{U} = U_d \times U_p$, which are then the total specified and unspecified spaces, respectively. This allows the definition of the pre-image in S of each $(u_d, u_p) \in \text{U}$ using f:

$$\Gamma(u_d, u_p) = \{(s_d, s_p) \in S_d \times S_p \mid f(s_d, u_d) = (s_p, u_p)\}. \tag{6.4}$$

If the set map f decoupled such that

$$\begin{aligned} f \quad : \quad & S_d \times U_d \rightarrow S_p \times U_p \\ & (s_d, u_d) \mapsto (s_p, u_p) = (f_1(s_d), f_2(u_d)), \end{aligned} \tag{6.5}$$

then $\Gamma(u_d, u_p) = S_d \times S_p$, the whole specified space. In this case, only part of the model (S_d and S_p) can be solved, as in Chapter 4, and no conclusions can be reached about the (U_d, U_p) aspects of the design.

Now the total space DPS $\times$ PPS has been partitioned into two sets, $\text{S} = S_d \times S_p$ on which preferences have been specified, and on $\text{U} = U_d \times U_p$ on which preferences remain unspecified. Given this formalization, an *induced preference* can be defined on the unspecified space U, to aid the designer in such specification. The restrictions made on S (*i.e.*, μ) can be induced onto U, to observe the induced restrictions on U.

The *Imprecision Transformation* is defined to be the formal transformation of the preferences as specified on S onto U. To define the transformation, consider the following informal argument, which will be made formal. Consider an unspecified parameter value $u \in \text{U}$. If a designer is restricted to be forced to exhibit u, and if two points in S can achieve this value u, then by the performance improvement principle (Assumption 2.1) the designer will choose to exhibit u with the point in S which the designer has more preference for. Finally, if the designer cannot achieve a value u with any $s \in \text{S}$, then that u must be given a rating of impossible.

This informal argument can be used to define the induced preference for any value $u \in \mathrm{U}$. Out of all the points in S that map to the particular u, define the induced preference as the supremum preference over its pre-image.

Definition 6.1 *Let S_d, U_d, S_p, U_p be sets. Let* DPS $= S_d \times U_d$, *and* PPS $= S_p \times U_p$. *Denote* $\mathrm{S} = S_d \times S_p$, *and let* $(s_d, s_p) = s \in \mathrm{S}$. *Denote* $\mathrm{U} = U_d \times U_p$, *and let* $(u_d, u_p) = u \in \mathrm{U}$. *Let* $f : \mathrm{DPS} \rightarrow \mathrm{PPS}$ *be a function, and* $\mu : \mathrm{S} \rightarrow [0,1]$ *be a preference. The* induced preference *on* U *is*

$$\nu(u) = \sup \{\mu(s) \mid s \in \Gamma(u)\}$$

and the convention is adopted that $\nu(u) = 0$ *if* $\Gamma(u) = \emptyset$.

This definition implies that the induced preference ν need not be continuous, even when in the situation where the DPS and PPS are topological spaces with continuous preferences and f is a continuous map. Also, if the set map is decoupled into f_1 and f_2 as in Equation 6.5, then the induced preference from S onto U is one for all $u \in U$: there is no relation between the specified aspects of the design and the unspecified aspect, and so one cannot make statements about the unspecified aspects. Any value $u \in \mathrm{U}$ will suffice at this point in the design process.

Graphically, the proposed computational model of a design process is depicted in Figure 6-1, called the *Method of Imprecision*. At the start of the process, the designer suggests possible solutions. These are subsequently transformed into formal candidate solution models upon which computations can be performed. Each model therefore includes a DPS and PPS as discussed in Chapter 2.

The designer then makes *a priori* estimates of values for any of the design and performance parameters desired. These preferences are then induced onto the remaining parameters, to observe the *a priori* restrictions on the remaining parameters.

After having made the induced preference calculations, the designer can observe the imprecise performance achievable and proceed to judge the candidates. This process can even continue when the designer has specified preliminary preferences on all the parameters. Given any parameter, the induced preference from the other parameters can be calculated,

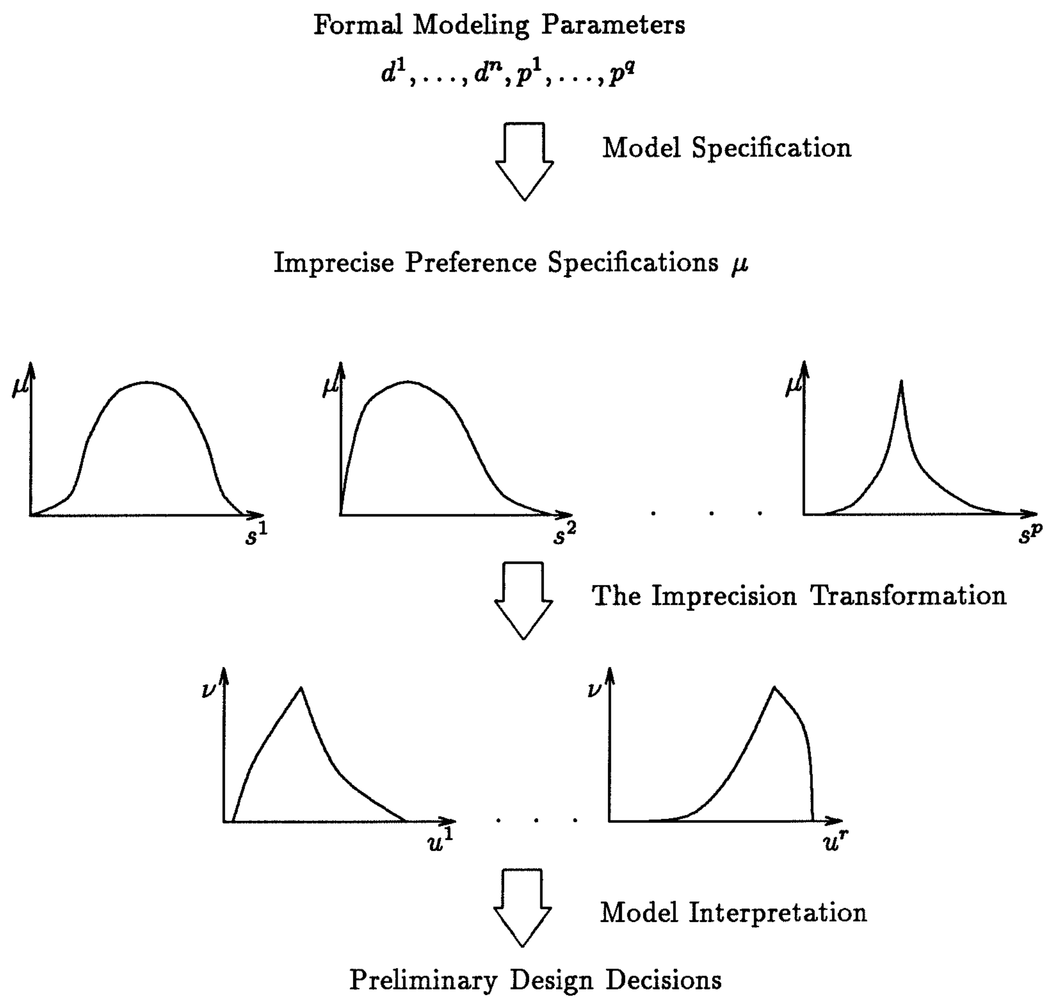


Figure 6-1: Transforming imprecision from S to U.

and the ν and μ can be compared.

Figure 6-1 presents a preliminary design process as if it could be carried out in a forward manner. It is intended that there will be *informal* iteration among the steps presented. Observations in the achievable performance will induce changes in the specified preferences. Thus, the design process characterized is both formal and informal. The method of imprecision, as presented here, allows the work to be split between informal designer interpretations and formal computations. This is of great benefit, for example, in that the formal computations can be performed by computer. Further, by definition, the informal interpretation cannot.

Specifications on the PPS

It is possible that the Method of Imprecision can be calculated in a simpler (more restricted) framework, and that is when preferences are initially specified only on the performance parameters. That is, suppose the designer is considering the restrictions imposed by performance specifications. The designer may wish to observe the possible design configurations allowed that satisfy specifications of preference on performance. For example, given a preference specification for cost (μ has been specified on a performance parameter reflecting the designer's interpretation of cost), the formal design configurations that can be built at the preferred expense can be computed.

In this case, formally the method of imprecision is reduced to the specified space S being the PPS, and the unspecified space U being the DPS (thus DPS = U = U_d, PPS = S = S_p, and $S_d = U_p = \emptyset$ in Definition 6.1). The induced preference definition (Definition 6.1) is more easily expressed here as:

Definition 6.2 *Let D and P be sets. Let $f : D \rightarrow P$ be a function, and $\mu : P \rightarrow [0,1]$ be a preference. The* induced preference *on D is*

$$\nu(d) = \mu \circ f(d).$$

Notice, therefore, the imprecision transformation reduces to composition when the specifications are made on the PPS (PPS = P in the definition), and are to be induced onto the

DPS (DPS $= D$ in the definition).

Specifications on the DPS

Though the last section demonstrates a clear simplification of the imprecision transformation, it is not the typical way in which a design process proceeds. Typically in engineering design, the designer has exclusive choice over design parameter values in the model, since the designer conceptualized the DPS. The designer chooses these values based on engineering judgment. The desired performance parameter values, on the other hand, can be reflections of requirements needed to be satisfied by the designer. The target values for these parameters are dictated by a customer. A comparison between the designer's estimates (on the DPS) and the customer requirements is desired. For example, for each configuration considered (to the degree of preference specified by μ), the cost can be observed, and then compared with the customer's desired cost.

In this case, formally the Method of Imprecision is reduced to the specified space S being the DPS, and the unspecified space U being the PPS (thus DPS = S = S_d, PPS = U = U_p, and $U_d = S_p = \emptyset$ in Definition 6.1). The induced preference definition (Definition 6.1) is more easily expressed here as:

Definition 6.3 *Let D and P be sets. Let $f : D \to P$ be a function, and $\mu : D \to [0,1]$ be a preference. The* induced preference *on P is*

$$\nu(p) = \sup\{\mu(d) \mid d \in D, f(d) = p\}$$

and the convention is adopted that $\nu(p) = 0$ if $f^{-1}(p) = \emptyset$.

In this simplification of Definition 6.1, the pre-image Γ (Equation 6.4) is the pre-image of f; *i.e.*, D is the DPS, and P is the PPS. Since this calculation is performed over the *entire set D* of possible points (design configurations), the designer is thereby allowed to make performance evaluations of every considered candidate configuration, rather than the usual case of one at a time evaluations. This will be made clear with the next example.

6.2.1 Maps of Real Parameters

The last section defined induced preference from the DPS to the PPS, which is a typical engineering design calculation: evaluating performance at various design configurations. A specific case of this is now considered by adding further structure to the spaces, namely restricting to real parameters.

Consider when the DPS $\simeq \mathbb{R}^n$, PPS $\simeq \mathbb{R}^m$, and each design parameter preference is specified independently. Such examples are common in engineering design, and so justifies particular development. Then, for each performance parameter p^j, the induced preference from the DPS can be computed. $\mu(d)$ is calculated by combining the individual preferences specified on the DPS by $\mu(d) = \mathcal{P}(\mu_1, \ldots, \mu_n)$. Thus, Definition 6.3 becomes:

Definition 6.4 *Let DPS* $\simeq \mathbb{R}^n$ *and PPS* $\simeq \mathbb{R}^m$, *and consider the* j^{th} *projection on the PPS, denoted* $\mathrm{P}_j \simeq \mathbb{R}$. *Let* f_j *represent this map,* $f_j : DPS \rightarrow \mathrm{P}_j$. *Let* μ_k *be the preference for the projection of the DPS onto* X_k. *The* preference induced by f_j *on* P_j *is the preference given by*

$$\nu(p^j) = \sup \left\{ \mathcal{P}\left\{\mu_1, \ldots, \mu_n\right\} \mid d^1, \ldots, d^n \in \mathbb{R}^n : p^j = f_j(d^1, \ldots, d^n) \right\}$$

and $\nu(p^j) = 0$ *if* $\{d^1, \ldots, d^n \mid p^j = f_j(d^1, \ldots, d^n)\} = \emptyset$.

6.2.2 Computation of Induced Preference

The definition just given presents difficulty in computation. A discrete numerical approach is necessary to meet the computational requirements for handling many design parameters. This section will discuss useful numerical techniques, which are based on performing interval calculations.

The Level Interval Algorithm

For a DPS $\simeq \mathbb{R}^n$, typically the design parameter preferences have compact support. Designers typically do not call for infinite values in their designs (though they might on performance parameters). Making this a formal restriction for the algorithms in this section, μ is assumed to have compact support, and this region is denoted $D_0 \subset$ DPS $\simeq \mathbb{R}^n$. Further, assume μ is convex over D_0, and μ is composed from n distinct preferences.

First, a preliminary definition is given of the level sets of preference over the DPS. For each $\alpha \in (0,1]$, define the set of all design parameters that have preference greater than or equal to α:

$$D_\alpha = \{d \in \text{DPS} \mid \mu(d) \geq \alpha\}. \tag{6.6}$$

D_α is defined for all α but zero, and so define D_0 as the support of μ. This definition is natural, in that $D_\alpha \to D_0$ as $\alpha \to 0$.

Now the *Level Interval Algorithm*, (LIA), as presented by Dong and Wong [39] (there called the "Fuzzy Weighted Average" algorithm), or as presented by Dong and Shah [38] (there called the "Vertex Method"), outlines a simple and efficient algorithm for calculating ν. A condensed version of the algorithm is presented below.

Algorithm 6.1 *For n real imprecise design parameters, consider a performance parameter represented by the mapping*

$$\begin{aligned} p \ : \ & \mathbb{R}^n \to \mathbb{R} \\ & \vec{d} \mapsto f(d^1, \ldots, d^n) \end{aligned}$$

Let $\nu(p)$ be the induced preference of the mapping. The following steps lead to the solution of ν.

1. For each d^i, discretize the preference function into a number of α values, $\alpha_1, \ldots, \alpha_M$, where M is the number of steps in the discretization.

2. Determine the intervals for each parameter $d^i, i = 1, \ldots, n$ at each α−cut, $\alpha_k, k = 1, \ldots, M$.

3. Using one end point from each of the n intervals for each α_k, combine the end points into an *n-ary* array such that 2^n distinct permutations exist for the array.

4. For each of the 2^n permutations, determine $p_l = f(d^1, \ldots, d^n), l = 1, \ldots, 2^n$. The resultant interval for the α−cut, α_k, is then given by

$$P_{\alpha_k} = [\min(p_l), \max(p_l)].$$

where P_α is the alpha cut of ν, as calculated by this algorithm, on the PPS.

It has been proven the algorithm correctly approximates ν when $\mathcal{P} = min$, and f is continuous monotonic [102]. Thus, the algorithm has, so far, only been proven for an equally weighted non-compensating design strategy.

Example 4: Design of a Truss

Consider the design of a structural truss, as shown in Figure 6-2. This example shall be denoted "Example 4." The truss is parameterized by design parameters w, t, l, W, E, where w and t are the width and thickness of the truss members, l is the length of the horizontal member, W is the applied load, and E is the elastic modulus of the material. Thus the DPS is a subset of $\mathbb{R}^5$. Imprecision in this model reflects that, at the start of a design process, the designer does not know what parameter values are desired to be used. For the purposes of this illustration, it is assumed that all these design parameters are crisp (only one value preferred), except for w and t, which the designer has given imprecise specifications for, as graphed in Figure 6-3. Thus, the DPS is simplified to $\mathbb{R}^2$, which permits the DPS to be shown on a plane.

Given a configuration (as shown in Figure 6-2), typically a designer performs calculations to rate different values of the design parameters. In this example, a typical performance parameter might be maximum bending stress in the horizontal bar. In terms of w and t with the remaining parameters fixed, this is given by

$$\begin{aligned} \sigma \ &: \ \mathbb{R}^2 \rightarrow \mathbb{R} \\ &(w,t) \mapsto \frac{2l(W + \frac{\rho g w t l}{6})}{wt^2}. \end{aligned} \tag{6.7}$$

The remaining parameter values are shown in Table 6.1.

Now given these specifications, the designer can perform calculations, such as observing how the preference specifications made on w and t induce preference on the performance parameter space. This will be calculated using the imprecision transformation, as reflected by Definition 6.4, with a non-compensating design strategy.

Observing that the performance parameter map is continuous monotonic over $D_0 \simeq [0.015, 0.105] \times [0.04, 0.12] \subset \mathbb{R}^2$, Algorithm 6.1 can be used to calculate $\nu(\sigma)$, and the results are graphed in Figure 6-4. Interpreting this figure, the designer determines that

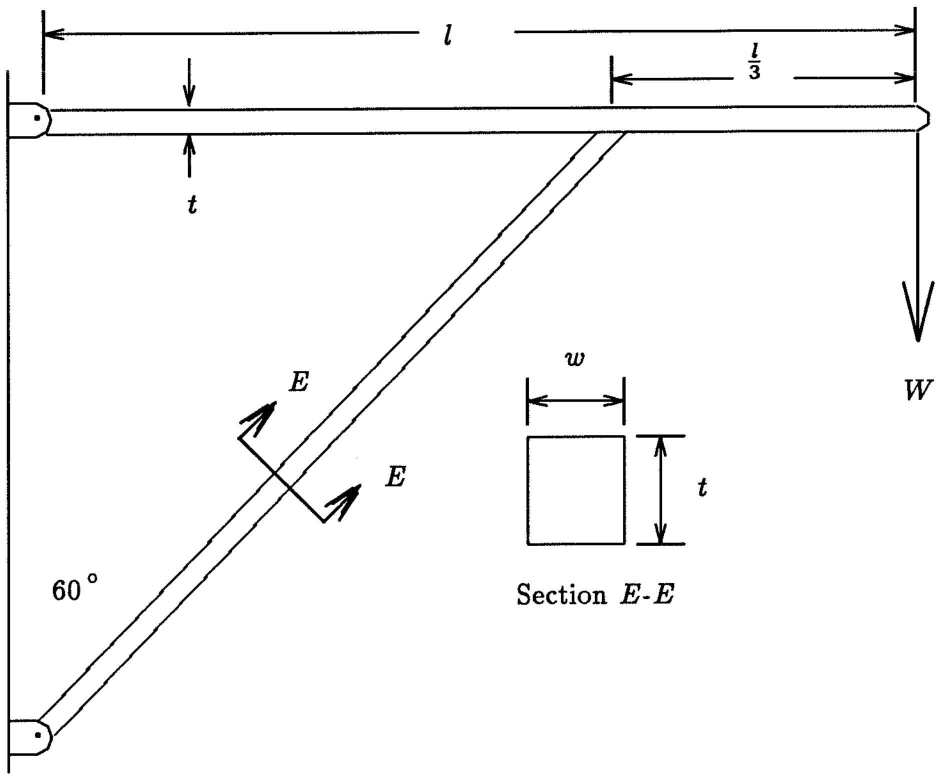


Figure 6-2: Example 4: structural truss.

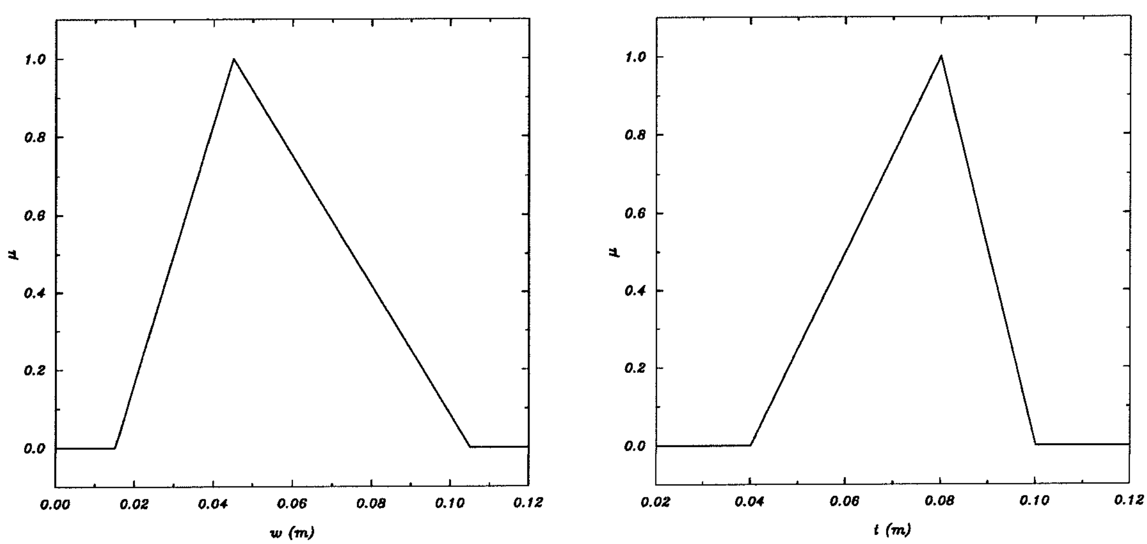


Figure 6-3: Example 4: specified design parameter preferences.

Table 6.1: Example 4: Constant Values.

parameter	Value
W	20 kN
l	4 m
g	9.8 $\frac{m}{s^2}$
ρ	7830 $\frac{kg}{m^3}$

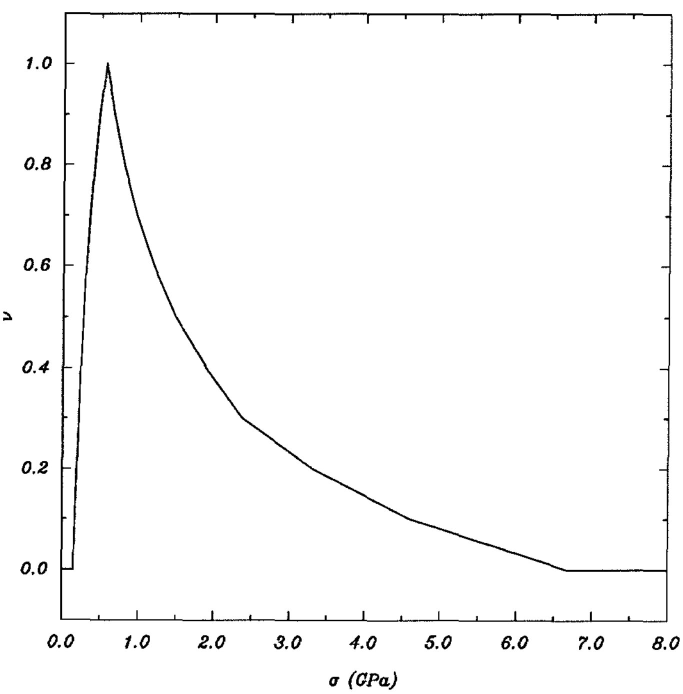


Figure 6-4: Example 4: induced preference ν.

if the most preferred point in the DPS could be used, $(w,t) = (0.045, 0.08)$ (expressed in meters), then the resulting stress would be 0.56 GPa. This example will be returned to later.

Extending LIA for Internal Extrema

The LIA algorithm presented above is valid only for real-valued functions f that do not include internal extrema within D_0. This is true due to only the endpoints (at a given α-cut) of the input parameters d^i, $i \in \{1,\ldots,n\}$ being used in the computation. An extension of the LIA algorithm to determine the correct bounds P_{α_k} is now presented for a given α-cut α_k with the following algorithm:

Algorithm 6.2 *For n real imprecise design parameters, consider a performance parameter represented by the mapping*

$$\begin{aligned} p \; : \; & \mathbb{R}^n \to \mathbb{R} \\ & \vec{d} \mapsto f(d^1,\ldots,d^n) \end{aligned}$$

Let $\nu(p)$ be the induced preference of the mapping. The following steps lead to the solution of ν.

1. For each α-cut α_k, determine if an internal extrema exists for the α-cut intervals of d^i, $i \in \{1,\ldots,n\}$, $p = f(d^1,\ldots,d^n)$. This may be accomplished by either analytically or numerically solving
$$\frac{\partial f}{\partial d^i} = 0$$
for each d^i.

2. Denote D_α the alpha cut of μ on the DPS, and P_α the alpha cut of ν on the PPS. Denoting the extrema by ξ_l, and denoting the values of the d^i that make up a given ξ_l by ε_l^i, if every ε_l^i lies within D_α, calculate ξ_l.

3. If every ε_l^i does not lie within the α-cut, let q index the set of those that do not, ε_l^q, and m index the set of those that do, ε_l^m. Denote the extrema within D_α as $f(\varepsilon_l^i) = \xi_l$. Denote the extrema on the boundary of the D_α caused by the extrema outside D_α (but within D_0) as $\xi_l = f(\varepsilon_{d^m,d^r})$, where d^m index the design parameters such that

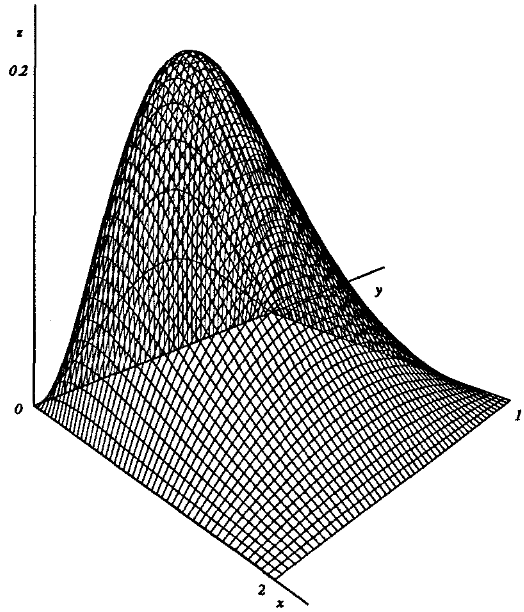


Figure 6-5: The map $f(x, y) = \frac{xy(x-2)(y-1)}{(x+\frac{1}{2})^2+(y-\frac{1}{2})^2}$.

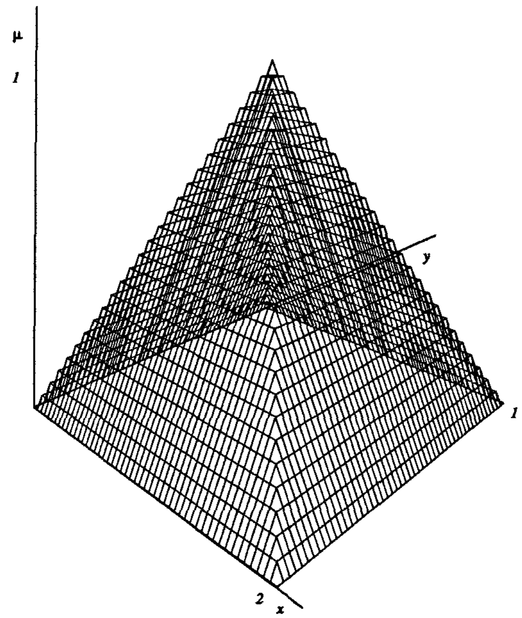
ε_{d^m} are within the α-cut, and d^r index the design parameters which are not within the α-cut, and use the values of d^r as the extrema of the α-cut on d^r.

4. If the above condition is true, compare the calculated extrema ξ_l with P_{α_j} from the LIA algorithm such that

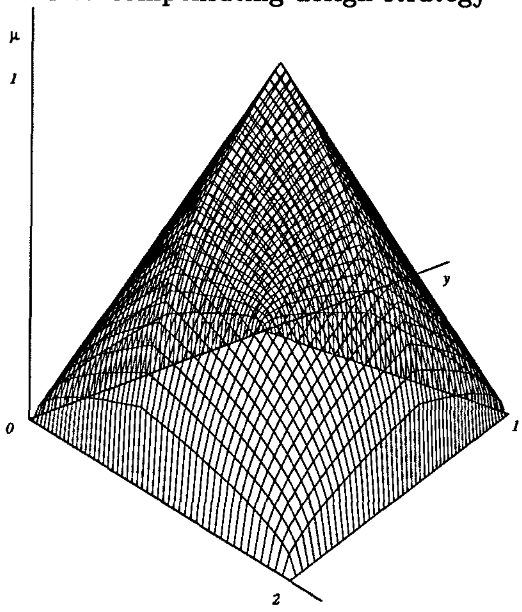
$$P_{\alpha_k} = [\min(p_{\alpha_k}, \xi_l), \max(p_{\alpha_k}, \xi_l)]$$

for all l.

The above algorithm correctly approximates ν when $\mathcal{P} = \min$, and f is continuous with only a finite number of extrema, since clearly any continuous function has its extrema over its critical points (the boundary and internal extrema), and the boundary of any D_α is the Cartesian product of the component intervals with a non-compensating design strategy.



Non-compensating design strategy



Compensating design strategy

Figure 6-6: Specified preferences $\mu(x, y)$.

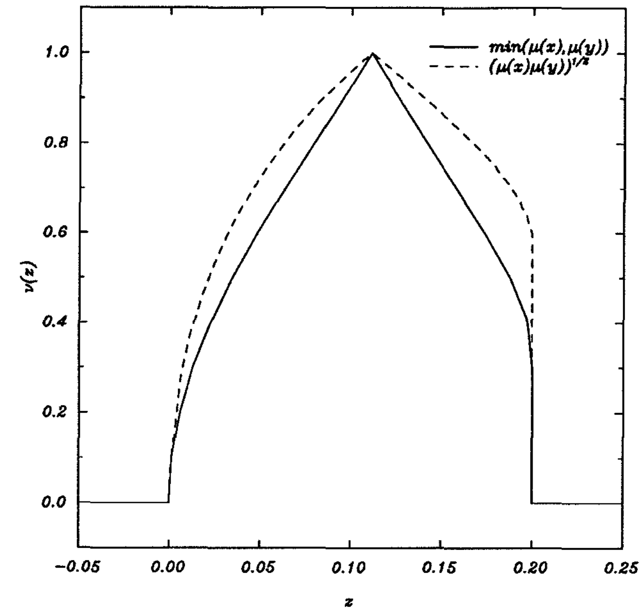


Figure 6-7: Induced preference ν.

As an example, consider the map

$$\begin{aligned} z \ &: \ \mathbb{R}^2 \rightarrow \mathbb{R} \\ &(x, y) \mapsto f(x, y) = \frac{xy(x-2)(y-1)}{(x+\frac{1}{2})^2+(y-\frac{1}{2})^2} \end{aligned} \tag{6.8}$$

on $D_0 = [0, 2] \times [0, 1]$, with triangular preference functions over x and y. The function f is graphed in Figure 6-5. The preference surfaces $\mu(x, y)$ are graphed in Figure 6-6 for the non-compensating and compensating equally weighted strategies.

This example has a bounded extrema point within D_0. Applying the two different strategies to determine a preference for a point in the DPS, this defines two induced preferences for any value z (applying Definition 6.4), depending on the strategy. The induced preferences $\nu(z)$ are graphed in Figure 6-7 for the two strategies. The non-compensating design strategy is the result of applying the bounded extremum LIA algorithm. It is not the same as the compensating design strategy result, calculated via the imprecision transformation.

Finally, note the bounded extremum LIA algorithm can be used for single design parameter problems regardless of the strategy, since in this case the issue of a strategy for combining design parameters is void; there is only one design parameter.

6.2.3 Anomalies in Imprecise Calculations

The algorithms presented above provide a basis for computing with sets of real imprecise parameters in preliminary engineering design. Even though the algorithms can be applied for continuous functions, limitations apply (as discussed above): the preference functions must satisfy the normality and convexity conditions, the performance mapping and preference functions must be continuous over D_0, and no singularities of the functions can occur within D_0. This section will now consider cases when one or more of these conditions are violated.

Discontinuous Preference Functions

Consider

$$\begin{aligned} z \ : \ & \mathbb{R} \rightarrow \mathbb{R} \\ & x \mapsto f(x) = x^3 - x \end{aligned} \tag{6.9}$$

(z is to be interpreted as a single performance parameter of a single design parameter x), whose graph is shown in Figure 6-8.

Let x be equipped with a triangular preference function $\mu(x)$ over $[-1, 1] \subset \mathbb{R}$. The LIA algorithm of the previous section cannot be applied, f contains extremum within the support of μ. Instead, Algorithm 6.2 must be applied.

Substituting $\mu(x)$ and f into the extended LIA algorithm for functions with bounded extrema, Algorithm 6.2, one obtains the resulting induced preference for z, graphed in Figure 6-9. Jump discontinuities in preference occur at $z = \pm 0.385$. These result from the local extremum in the map f, shown in Figure 6-8 ($x = \pm 0.577$). At z values just above $z = 0.385$, only points in the neighborhood of $x = 1.15$ are in the pre-image of these z; the map is one-to-one. But at z values just below $z = 0.577$, the pre-image contains points in the neighborhood of both $x = -0.577$ and $x = 1.15$. Values of z between the local extremum in Figure 6-8 ($z \in [-0.385, 0.385]$) have 3 points x in their pre-image. Points in the neighborhood of $x = -0.577$ have higher preference than points in the neighborhood of $x = 1.15$, and hence at $z = 0.385$ there is a jump discontinuity in preference.

This type of discontinuity in induced preference is easily understood; it arises from the multiplicity of points in the map's pre-image. The multiplicity here is finite, and can be dealt with. In particular, this is a demonstration of when Algorithm 6.2 must be applied. Real concerns arise, however, when the induced preferences exhibit unboundedness, either in support or in the number of discontinuities. Representation and calculation of the imprecise preferences then becomes difficult, as will be demonstrated below.

Unbounded Preference Functions

Consider a real design parameter x that the designer desires to be "about 1/4, and possibly negative," *i.e.*, let $\mu(x)$ be a triangular preference, where $\mu(\frac{1}{4}) = 1$, and $\mu(x) = 0$ outside $\{x \in \mathsf{R} \mid x \in (\frac{1}{4} - s, \frac{1}{4} + s)\}$, and s will be gradually increased to observe the variation in calculation results. The graph of μ is shown in Figure 6-10 for various values of s. It is expected that such an imprecise number will become ill-defined with inversion (when s becomes large enough to encompass zero), since the inverse of points in the neighborhood of zero become unbounded.

The effect of the inverse map (extended by defining $f(0) = 0$)

$$\begin{aligned} z \ &: \ \mathsf{R} \rightarrow \mathsf{R} \\ &x \mapsto f(x) = 1/x \end{aligned} \tag{6.10}$$

was observed as s was increased. Substituting $\mu(x)$ and the map f into Definition 6.4, one obtains the induced preference of any z as

$$\nu(z) = \mu(x), \text{ where } x : x = 1/z. \tag{6.11}$$

The results are graphed in Figure 6-11. As s increases, the induced preference for z begins to extend to include non-zero preference for unbounded values; all calculations are performed with the nominal desired value for x remaining at 1/4. This is a case of a map that transforms a well-formed imprecise number and produces an induced preference with unbounded support. This occurs due to D_0 (the support of μ) including 0, a singularity of the map f. Hence the induced preference includes $f(\epsilon) = 1/\epsilon$ approaches ∞ as ϵ be-

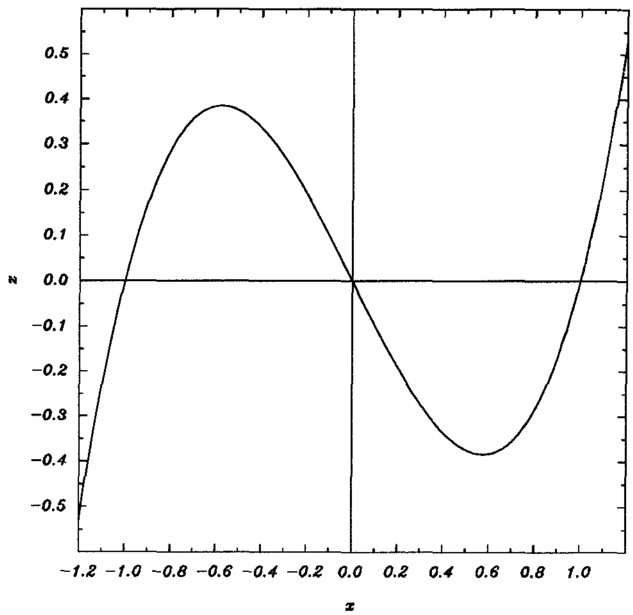


Figure 6-8: The cubic map $f(x) = x^3 - x$.

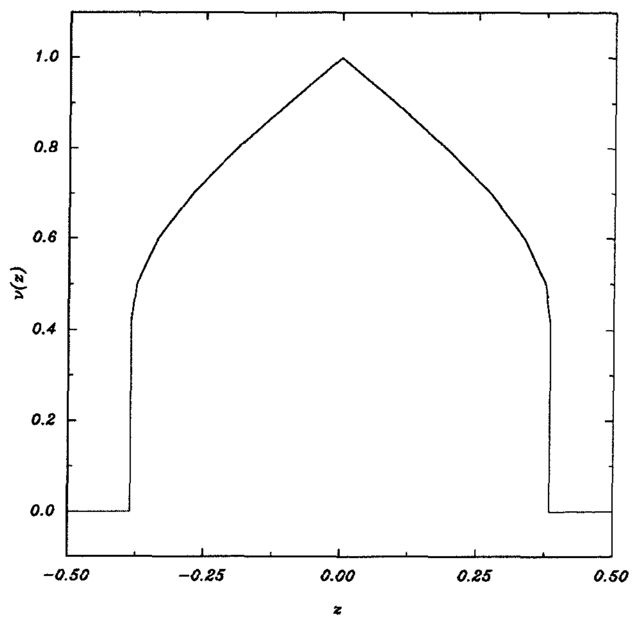


Figure 6-9: Induced preference ν.

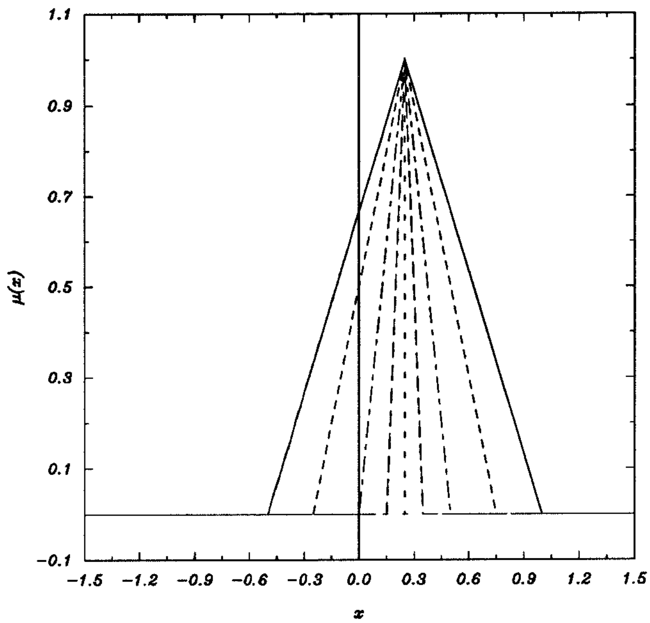


Figure 6-10: Preferences $\mu(x)$ around zero.

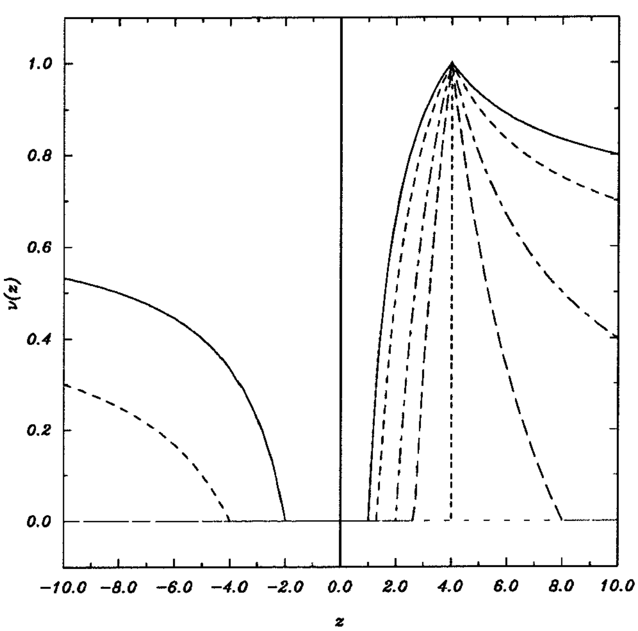


Figure 6-11: Induced preferences ν.

comes small. When the input preference $\mu(x)$ includes both the small positive and negative points (the neighborhood of zero), the induced preference function $\mu(z)$ has support over both positive and negative values in the "neighborhood" of infinity. Therefore the induced preference function will appear as graphed in Figure 6-11.

The LIA algorithm will fail, of course, since zero is a singularity of f, and so cannot occur in D_0. This result seems to indicate that using induced preferences to represent designer uncertainty will sometimes fail. But consider what the result indicates when the preference function is interpreted as the degree of designer preference. The unbounded results on z are due to the preferences stated on x. The problem has been incorrectly formulated, and must be reformulated to hedge the preferences on x away from the singular point zero, just as the designer would have to do if imprecision were not used. This method warns the designer of the problem, whereas using crisp or single valued calculations (in the example, just using $x = 1/4$) would not.

Singular Points and Preference Functions

Consider

$$\begin{aligned} z \ &: \ \mathbb{R} \rightarrow [-1,1] \subset \mathbb{R} \\ x &\mapsto f(x) = \begin{cases} \sin(1/x) & x \neq 0 \\ 0 & x = 0. \end{cases} \end{aligned} \tag{6.12}$$

This function has a non-removable singularity at $x = 0$. As $x \rightarrow 0$, f oscillates an unbounded number of times between -1 and 1. Refer to Figure 6-12. The question then arises as to the induced preference behavior when the input preference μ includes in its support the singular point $x = 0$.

Let x be equipped with preference function $\mu(x)$, where $\mu(x) = 0$ outside $\{x \mid x \in (-\pi + s, \pi)\}$, $\mu(\frac{s}{2}) = 1$, and s is gradually decreased from $\frac{3}{2}\pi$ to 0 to observe the variation in calculation results. The graph of μ is shown in Figure 6-13 for various values of s. Substituting μ and f into Definition 6.4, one obtains the resulting induced preference for z

$$\nu(z) = \sup\{\mu(x) \mid x : z = \sin(1/x)\}. \tag{6.13}$$

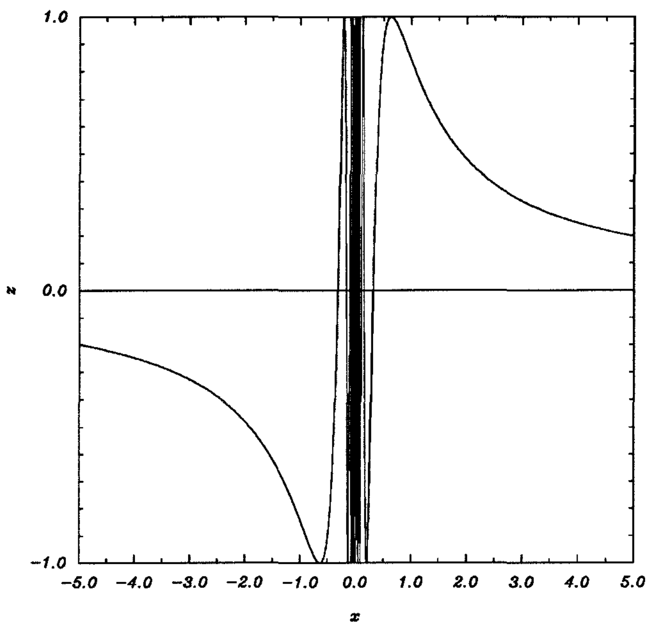


Figure 6-12: The map $f(x) = \sin(1/x)$.

Figure 6-14 graphs the corresponding induced preferences resulting from application of Equation 6.13 to various μ of Figure 6-13.

When the support for μ does not include the singular point 0 (*i.e.*, $s > \pi$), the induced preference remains perfectly well defined. When μ does include the singular point for any support range, it is expected that the induced preference would become not well defined, since, after all, the map f itself becomes oscillatory (not well-defined). But note that the induced preference remains well defined. This is due to the supremum in the imprecision transformation definition, and the selected preference μ. There is always a point x in each z value's pre-image such that the preference of that x is greater than the singular point preference. Since this supremum preference is greater than the singular point preference, the preference for all z are well defined. Hence the preference $\nu(z)$ is well defined when $x = 0$ does not have the peak preference among all x. When $x = 0$ has the peak preference, the *sup* definition must be explicitly used, since the point with maximum preference in the pre-image of any z does not exist, only a least upper bound exists. All values of z have in their pre-image points x arbitrarily close to zero. In this case, the peak preference of $x = 0$

would be the least upper bound of preference for all z, *i.e.*, $\nu(z) = 1\ \forall z$.

Calculation of the induced preference for $f(x) = \sin(1/x)$ poses difficulty. There are numerous and possibly an infinite number of internal extrema within the support of the design parameter (depending if the neighborhood of $x = 0$ is in the support). Hence the methods of Section 6.2.2 fail as posed. They must be extended using a limiting process. That is, the support of μ must be split into monotonic intervals of f between the zeros of the slope of f as discussed and solved in Section 6.2.2. The difference here is that there are an unbounded number of such intervals. These intervals, however, will become smaller in a limit as will their contribution to $\nu(z)$. The limiting process can be terminated with a convergence criteria. Alternatively, Equation 6.13 can be directly solved using a limit process as well.

Given that functions exist that exhibit such computational difficulty, one must have an ability to characterize them to know when the methods of Section 6.2.2 will fail. This problem's key feature which causes the interval methods to fail is the presence of the singularity of the function f in the support of μ. The same can be said of the previous example involving unbounded support ($f(x) = 1/x$). In both cases, the induced preference functions are defined and exist, but they exhibit difficulty in representation or evaluation. These effects are a direct result of the singularity of the function being within the support of the design parameter preference.

Note for $f(x) = \sin(1/x)$, the singularity requires particular care, in that $\nu(z)$ is well behaved, even though there is a singularity of f within μ. Any conclusions drawn by a designer when observing such a resulting induced preference for z may be misleading. The point is that failure to meet the criterion of using the interval algorithms of Section 6.2.2 indicate that the performance maps being considered (f) are incompatible with the specified design parameter preference functions (μ), and hence is a warning to the designer that the values desired for x should be re-considered.

Interpretability of Imprecision Results

Consider the possible existence of continuous maps which operate on well-formed design parameter preferences and with them create induced performance parameter preferences

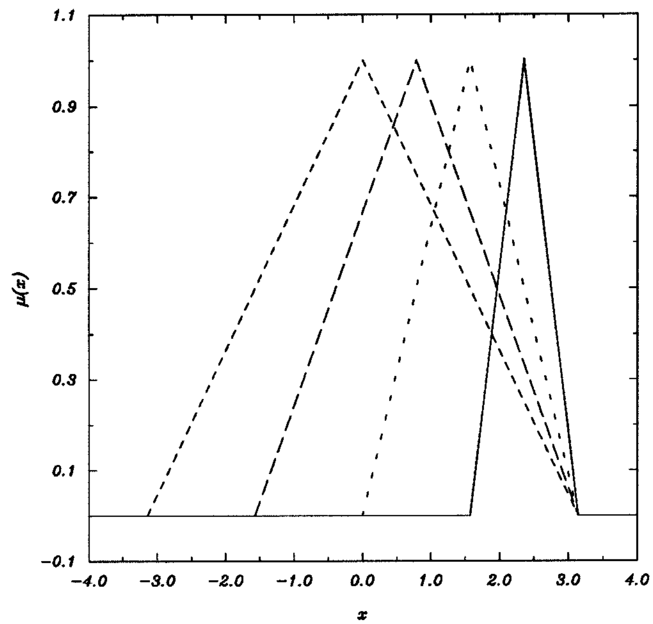


Figure 6-13: Preferences $\mu(x)$.

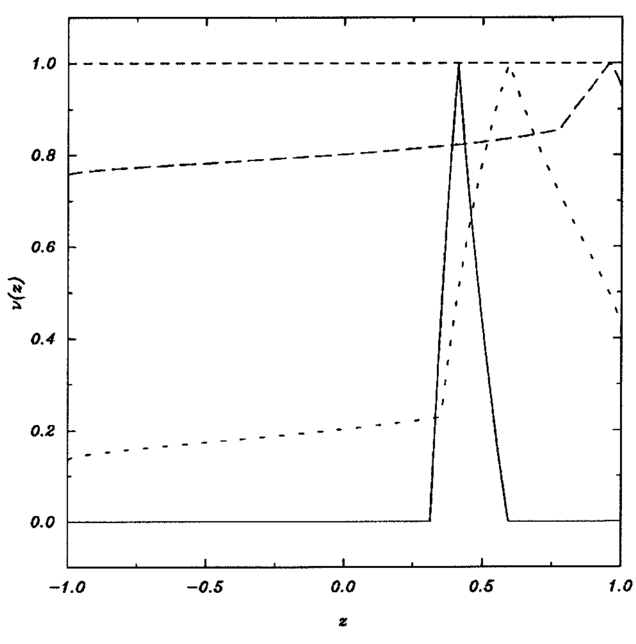


Figure 6-14: Induced preferences ν.

that oscillate in preference from 0 to 1 as the output z approaches a limit value $\overline{z}$. The existence of any such map would question the viability of the method of imprecision, since such a map would be an interpretable mapping from a parameter set where the preferences are known to a set where the induced preferences are un-interpretable. Via construction, it will be shown (though not formally proven) that this is only possible for maps that are infinitely multi-valued (have an infinite number of branches), but need not be singular. Any such maps that transform specified preferences into un-interpretable preferences are not due to the imprecision transformation (Definition 6.3), but are entirely due to multi-valued character of the function, and hence would exist whether the method of imprecision were used or not.

Such a map f must convert a well-formed, convex design parameter preference $\mu(x)$ into an induced preference of the form $\nu(z) = 1/2\sin(1/z) + 1/2$, or more simply behave like $\sin(1/z)$, *i.e.*, behave un-interpretable in the neighborhood of $\overline{z} = 0$. Denoting $z = f(x)$ and using the imprecision transformation (Definition 6.4), this produces $f(x) = 1/\arcsin(\mu(x))$. Now define $\mu(x)$ as a convex function over all real values with range $[0, 1]$, such as e^{-x^2}. These definitions construct

$$\begin{aligned} z \ &: \ \mathbb{R} \rightarrow \mathbb{R} \\ &x \mapsto f(x) = 1/\arcsin(e^{-x^2}) \end{aligned} \tag{6.14}$$

based on $\nu(z)$ and $\mu(x)$.

Using different functions that exhibit the same behavior as the chosen $\nu(z)$ and $\mu(x)$ will construct a similar f, though it may not be as easily expressible. The created f is defined over all x and contains no singularities. It is, however, multi-valued. Refer to Figure 6-15. Take f to be all the branches. Then the pre-image of each value z has either zero or two points x.

To observe the effects of f on an imprecise number x, consider μ where $\mu(x) = 0$ outside $(-1, 1)$, $\mu(0) = 1$, and $\mu(x)$ smoothly increases to the peak at 0 between -1 and 1. Substituting μ and the equation for f into the imprecision transformation, the resulting ν is graphed in Figure 6-16.

Note that for z values in the neighborhood of zero, the induced preference becomes

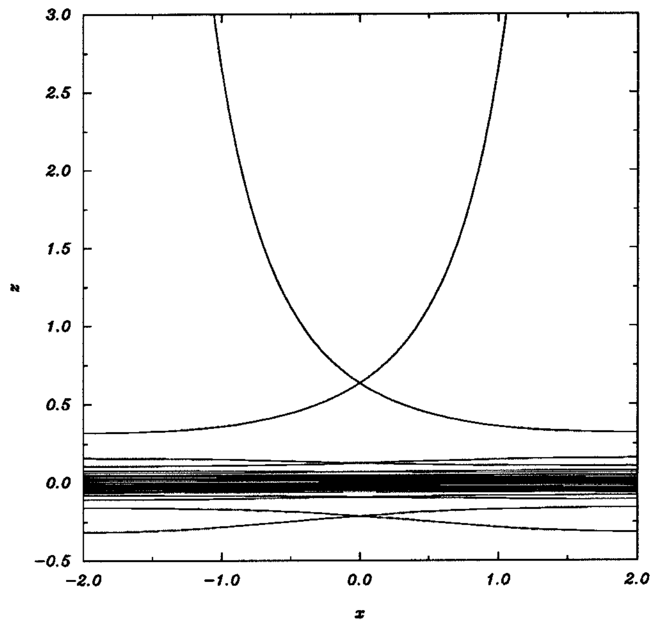


Figure 6-15: The map $f(x) = \frac{1}{\arcsin\left(e^{-x^2}\right)}$.

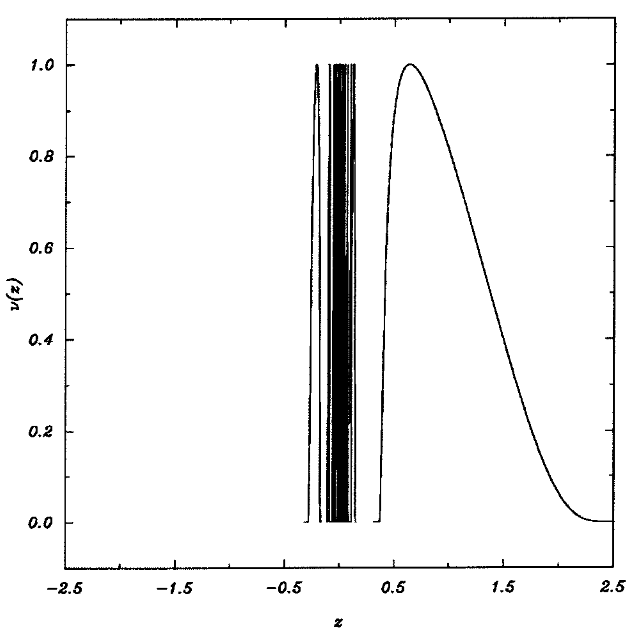


Figure 6-16: Induced preference ν.

arbitrary, from 0 through 1 in preference, as was desired to exhibit. That is, moving a δ amount in the neighborhood of $z = 0$ will result in large changes in ν, from zero to one. Also note the intervals on z where the preference function is not mapped from any x (for example, all z values between $-\infty$ and $\sin(1/\pi)$). These z values require complex x; there are no real x in the pre-image of these z. The number of such intervals in a neighborhood of z becomes unbounded as z approaches zero (and the length of the each interval approaches zero length). These intervals arise since f is not a surjection; *i.e.*, $Z = f(\mathbb{R}) \not\simeq \mathbb{R}$, but $Z \subset \mathbb{R}$. According to the interpretation of preference, $\nu \simeq 0$ for the values of z having no real points x in the pre-image. Therefore, defining $\nu = 0$ for such points, this defines $\nu\ \forall z$, as reflected in Definition 6.4.

These results suggest that a preference function could become un-interpretable, since this preference function became un-interpretable as $z \to 0$. But this was entirely due to the unbounded number of branches considered in f: f is not a function. For any particular branch, however, the mapping is well defined. This result (infinitely many choices as $z \to 0$) exists using crisp numbers as well as imprecise numbers.

Note f was constructed based on the desired induced preference $\nu(z)$ (being un-interpretable at a value). The only way to exhibit this behavior was with a multi-valued map, where the multi-valued character arose as a direct consequence of the desired un-interpretability. Thus the conclusion: only multi-valued functions can lead to un-interpretability, entirely due to the indecision on the branch that one is using in the function f itself, not in the imprecise mathematics. Once a branch is selected, the problem reduces to a simple case that the methods of Section 6.2.2 can accommodate. Replacing the specific *sine* used for ν by any general periodic function and replacing the specific μ by a general convex imprecise number in the construction demonstrates the conclusion for the general case (though this is not a formal proof). Overall, if the performance expression is interpretable, then the imprecision method will also produce interpretable results.

Related Work

Consider when the initial preferences are specified exclusively on the DPS, S = DPS, and each design parameter has an associated independent preference function. Further, assume these are combined with a non-compensating design strategy, and so $\mathcal{P}$ is the *min*. With these restrictions, Definition 6.3 reduces to Zadeh's extension principle, from fuzzy set theory [200]. Wood, Antonsson, and Beck have shown the extension principle is a well suited method for imprecise design calculations [190], rather than, for example, using the probability mathematics. The extension principle preserves peak preferences on the performance parameters with peak preferences on the design parameters, a required feature for the transformation.

On the other hand, notice that when S = PPS, or the preferences are specified on the PPS, then the imprecision transformation simply reduces to composition: $\nu(d) = \mu \circ f(d)$. This is, for example, as utility theory [58, 129] formalizes problems. A preference is composed on the performances, and the composition is optimized to determine the optimal d, similar to Chapter 4.

Other authors have considered computational methods for applying the extension principle to fuzzy numbers (real design parameters with continuous normalized preference functions with compact support). Kaufmann and Gupta [83] propose a method for analytical arithmetic with fuzzy numbers. They introduce the properties of fuzzy numbers, along with an α-cut formulation of the operations on number sets.

In [41, 48, 50], Dubois and Prade review the correctness of using α-cut calculations for a non-compensating design strategy. This has been shown in earlier work by Negoiţă and Ralescu [102], and also by Nguyen [103] for 2 variable problems, and in [105] by Novák. It is presented clearly in a recent work by Buckley and Qu [25], and is reviewed in Appendix B, as required for the invariance proofs to be presented in Section 6.3.

Dong and Wong [39], as described above, present a computational algorithm for implementing the extension principle. The application domain for this algorithm emphasizes purely algebraic expressions, with a particular focus on fuzzy weighted averages. In [39], Dong and Wong also review the non-linear programming technique developed by Baas and Kwakernaak [10], an analytical procedure for L-R membership functions due to Dubois and Prade [42], an investigation of fuzzy number algebraic properties by Mizumoto and Tanaka [97], and an investigation of fuzzy algebra using interval operations by Nakamura [101]. These techniques and investigations provide further methods for evaluating induced preference.

Dong and Shah [38] also present a computational algorithm for implementing the extension principle. The application domain for this algorithm emphasizes purely algebraic expressions. They also introduce an algorithm for computing the extension principle with functions with bounded extrema, similar to Algorithm 6.2. The algorithm as presented in [38], however, will compute incorrect bounds in dimensions higher than 1. The problem is that one must consider combinations of the components of the vertex points with the components of the singularity points, as is done in Algorithm 6.2.

6.2.4 Coupled Equations

The previous section considered when the PPS could be stated in terms of $\mathbb{R}\times\cdots\times\mathbb{R}$, where each projection p^j can be considered independently of the others. Consider now when the performance parameters are coupled.

Let $\mathfrak{p}$ denote an element of a projection from the PPS to a coupled subset of the PPS: $\mathfrak{p}$ has coupled component parameters. Then Definition 6.3 becomes, on the subspace:

$$\nu(\mathfrak{p}) = \begin{cases} \sup\{\mu(d) \mid d \in \text{DPS} : \mathfrak{p} = \pi_{\mathfrak{p}} \circ f(d)\} \\ 0 \quad \text{if } \{d \in \text{DPS} \mid \mathfrak{p} = \pi_{\mathfrak{p}} \circ f(d)\} = \emptyset \end{cases} \tag{6.15}$$

where d are values of the imprecise design parameters, $\pi_{\mathfrak{p}}$ is the projection, $f(d)$ is the performance parameter map, and μ is the preference for the design parameter value d used.

For example, consider a linear coupled system of parameters described by

$$\begin{bmatrix} d^{11} & d^{12} \\ d^{21} & d^{22} \end{bmatrix} \begin{Bmatrix} p^1 \\ p^2 \end{Bmatrix} = \begin{Bmatrix} d^3 \\ d^4 \end{Bmatrix}. \tag{6.16}$$

In this example, given preferences specified on the d^i, one cannot graph the induced preferences for either p^j without holding the other fixed. Both must be considered simultaneously to calculate an induced preference $\nu(p^1, p^2)$ on any vector (p^1, p^2). For example, consider triangular preferences on each d^i as specified by Table 6.2. Consider a non-compensating design strategy. Then the induced preference on (p^1, p^2) is depicted as level sets in Figure 6-17.

As can be seen by Figure 6-17, even simple coupled systems can induce complex preferences. Such computations may be difficult to perform. With a non-compensating design strategy,[1] however, interval methods [99] can be used to calculate level sets of ν. This difficultly is a computational concern; however, the induced preference for such systems is clearly defined.

6.2.5 Differential Equations

In this section some of the basic notions of induced preference are developed for cases when the DPS and PPS have the structure of a differentiable manifold.[2] Specifically, the preference induced by the Lie derivative of a function along a vector field, and the preference induced by the flow of a vector field, possibly depending on parameters, is defined. In many problems in physics and engineering, situations arise where it is necessary to perform calculations on manifolds, and the extension of induced preference to these situations is then needed.

The examples that will be presented in this section are meant to be illustrative. Given an application, it should be straightforward, in principle, to proceed using the methods outlined here.

Maps on Manifolds

The discussion in this section will be presented in the context of differentiable manifolds (all work will remain in the C^∞ category).

Let M be a smooth manifold (considering, for example, $M \simeq$ DPS). Then a preference on M is a map $\mu : M \to [0, 1]$. μ can be directly specified on the manifold, or may be constructed using a $\mathcal{P}$ on the components of a parameterization. No smoothness requirements are placed on μ. Indeed, in the examples to be presented, none of the preference functions are smooth, and some of them are not even continuous.

[1] Also when the geometric conditions to be discussed in Section 6.3 are satisfied between $\mathcal{P}$ and $\pi_p \circ f$.

[2] The results presented in this and the next two sections were developed with assistance from Mr. Andrew Lewis, currently at Caltech.

Table 6.2: Coupled Equation Example: Triangular μ Specifications.

d^i	$\alpha = 0$	$\alpha = 1$	$\alpha = 0$
11	0.5	1.0	1.5
12	1.0	2.0	3.0
21	-1.5	-1.0	-0.5
22	0.25	1.0	1.75
3	3.0	6.0	9.0
4	6.0	9.0	12.0

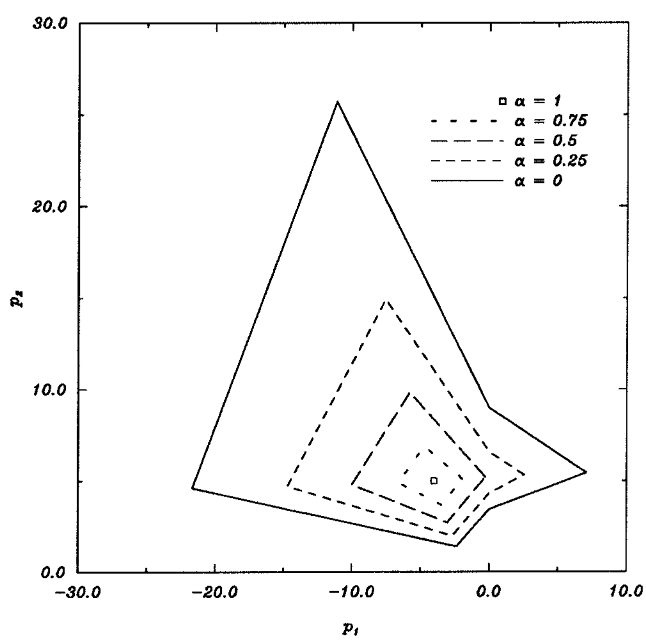


Figure 6-17: Induced preference ν.

If N is another manifold and if $f : M \to N$ is a smooth map (considering again, for example, PPS $\simeq N$), then the induced preference (Definition 6.3) becomes

Definition 6.5 *Let M and N be smooth manifolds, and $f : M \to N$. Let $\mu : M \to [0,1]$ be a preference. The* preference induced by f *is the preference on N given by*

$$\begin{aligned} \nu &: N \to [0,1] \\ & n \mapsto \sup\{\mu(m) \mid m \in M, \quad f(m) = n\} \end{aligned}$$

If $f^{-1}(n) = \emptyset$, then take $\nu(n) = 0$.

Observe that this definition is nothing more than Definition 6.3 when D and P are manifolds.

Example: Let $M = \mathbb{R}$ be equipped with the preference function whose graph is shown in Figure 6-8 (a simple triangular preference centered at zero). Define

$$\begin{aligned} z &: M \to N = \mathbb{R} \\ &: x \mapsto f_1(x) = x^3 - x. \end{aligned} \tag{6.17}$$

Figure 6-9 shows the preference induced on $\mathbb{R}$ by f_1. This illustrates the induced preference on values of the function f_1 given the preference on x as graphed in Figure 6-9.

Example: Let $M = \mathbb{S}^2 = \{\vec{x} \in \mathbb{R}^3 \mid \quad ||\vec{x}|| = 1\}$. If (x, y, z) are the standard coordinates on $\mathbb{R}^3$ then define a preference function on M by $\mu(x, y, z) = y^2 + z^2$. Thus $\mu = 1$ on the equator $x = 0$, and $\mu = 0$ at the poles $(1, 0, 0)$ and $(-1, 0, 0)$. Let f_2 be the function on M given by $f_2(x, y, z) = x^2 + y^2$. The range of f_2 is the interval $I = [0, 1]$. For any $\xi \in I$, $\nu_{f_2}(\xi) = 1$ since $f_2^{-1}(\xi)$ will contain at least one point (x, y, z) on M where $x = 0$. At this point $\mu(x, y, z) = 1$, and so when the supremum is computed, 1 will be the result. Thus $\nu_{f_2}(\xi) = 1$ when $\xi \in I$ and $\nu_{f_2}(\xi) = 0$ otherwise. This example will receive more attention in the next section.

Preference Induced by Lie Differentiation

In this section a definition is given for the preference induced by the derivative of a function along the integral curves of a vector field. The set of real-valued (C^∞, by hypothesis) functions on a manifold M will be denoted by $\mathcal{F}(M)$, and the set of vector fields on M

will be denoted by $\mathcal{X}(M)$. Thus, notice a temporary change of notation, to conform with traditional notation of vector fields. X now denotes a vector field, not a subset of the DPS as was used in previous sections. This will only be true in this section.

Let $X \in \mathcal{X}(M)$ and $f \in \mathcal{F}(M)$. The *Lie derivative* (see Lang [89]) of f along X will be written $\mathcal{L}_X f$. If $(d^1, \ldots, d^n)$ are coordinates in a chart for M, and if, in this chart, the vector field X is represented by $(X^1(d^1, \ldots, d^n), \ldots, X^n(d^1, \ldots, d^n))$ (*i.e.*, $X = X^i \partial/\partial d^i$, with summation over repeated indices), then

$$\mathcal{L}_X f = X^i \frac{\partial f}{\partial d^i} \tag{6.18}$$

with summation over repeated indices. Thus, for every $X \in \mathcal{X}(M)$, the Lie derivative assigns a function $\mathcal{L}_X f$ to every $f \in \mathcal{F}(M)$. Thus preferences can be computed for $\mathcal{L}_X f$ in the manner given in Definition 6.5.

Thus, the Lie derivative generates a new function, to which the induced preference can be defined. Equation 6.18 in conjunction with Definition 6.3 leads to the following definition.

Definition 6.6 *The* preference induced by f along X *is the preference induced on* $\mathbb{R}$ *by* $\mathcal{L}_X f$. *This will be denoted by*

$$\nu_{X,f}(\eta) = \sup\{\mu(m) \mid m \in M, \quad \mathcal{L}_X f(m) = \eta\}$$

where $\eta \in \mathbb{R}$. *If* $(\mathcal{L}_X f)^{-1}(\eta) = \emptyset$, *take* $\nu_{X,f}(\eta) = 0$.

As a special case of this definition, let $M = \mathbb{R}^n$, and let $(d^1, \ldots, d^n)$ be the standard coordinates for M. Then there are n distinguished vector fields on M given by $X_i = \partial/\partial d^i$, for $i = 1, \ldots, n$. For a function $f : \mathbb{R}^n \to \mathbb{R}$, $\nu_{X_i,f}$ will give the preference on $\mathbb{R}$ induced by the function $\partial f/\partial d^i$ on $\mathbb{R}^n$. In particular, if $n = 1$, then Definition 6.6 reduces to the definition of induced preference for differentiation on $\mathbb{R}$ as presented in [42] by Dubois and Prade for fuzzy sets (strategy issues are void in one dimension).

Thus, an induced preference can be determined for the case of manifolds using Lie differentiation. Recall, however, the overall objective in doing so. The designer wishes to observe the induced preference on a derivative of a map, as well as considering the induced preference on the map itself. One can consider these both as independent performance parameters, observe the induced preferences, and use the previous development to determine design performance. There is a problem, however, if such an approach is taken. The induced preference on the derivative is not independent of the induced preference on the function itself. These must be considered together.

So, given a function f and a vector field X on a manifold M, a preference on $\mathbb{R}^2$ can be defined that represents the preference induced by f along with the preference induced by f along X. More precisely, a preference is induced on $\mathbb{R}^2$ as follows.

$$\begin{aligned} \tilde{\nu}_{X,f} \quad &: \quad \mathbb{R}^2 \to [0,1] \\ &(\xi, \eta) \mapsto \sup\{\mu(m) \mid m \in M, \quad f(m) = \xi, \quad \mathcal{L}_X f(m) = \eta\} \end{aligned} \tag{6.19}$$

As usual, if $f^{-1}(\xi) \cap \mathcal{L}_X f^{-1}(\eta) = \emptyset$, then $\tilde{\nu}_{X,f}(\xi, \eta) = 0$. This is an example of Definition 6.5 where $N = \mathbb{R}^2$.

To illustrate the use of Equation 6.19, the previous examples of this section are treated again with the addition of differentiation along specified vector fields.

Example: Let $M = \mathbb{R}$ and define f_1 as before, Equation 6.17 ($f_1(x) = x^3 - x$). As a preference on M, take the function whose graph is depicted in Figure 6-8. Let $X_1 = \partial/\partial x$, where x is the standard coordinate for $\mathbb{R}$. A quick calculation gives $\mathcal{L}_{X_1} f_1 = 3x^2 - 1$. Figure 6-18 graphs the induced preference for the derivative. Recall Figure 6-9 graphs the induced preference for f_1. Figure 6-19 shows the preference on $\mathbb{R}^2$ induced by f_1 and $\mathcal{L}_{X_1} f_1$ as given by Equation 6.19. Observe that if the curve representing non-zero preference is projected onto the $(\xi, \tilde{\nu}_{X,f})$-plane, the preference induced by f as graphed in Figure 6-9 results, and if the same curve is projected onto the $(\eta, \tilde{\nu}_{X,f})$-plane, the preference induced by f along X results, as graphed in Figure 6-18. This observation is true in general, as can easily be seen by checking the definition.

Example: Let M, μ, and $f_2(x, y, z) = x^2 + y^2$, as was defined in the second example of this section. Define a vector field on $\mathbb{R}^3$ by $X_2 = y\partial/\partial x - x\partial/\partial y$. Thus X_2 generates a uniform rotation about the z-axis in $\mathbb{R}^3$. It is easily verified that X_2 leaves M invariant, and so defines a vector field on M by restriction, which is denoted by X_2. The integral curves of X_2 on M are shown in Figure 6-20. In Figure 6-21 the preference on $\mathbb{R}^2$ is graphed as

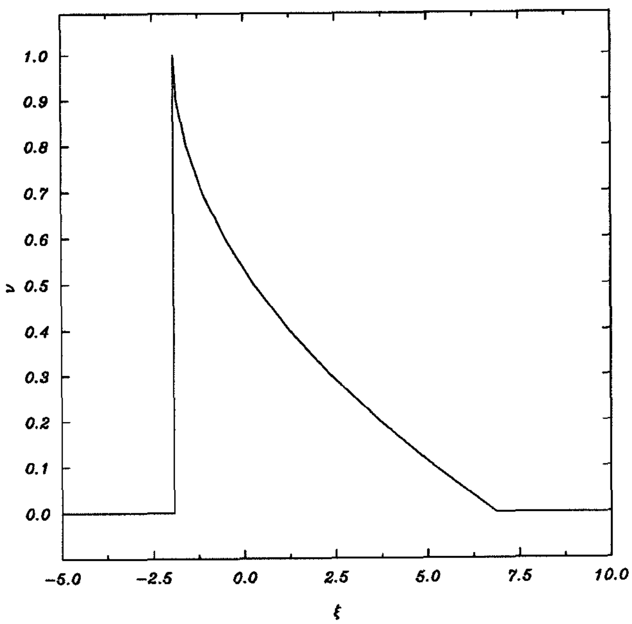


Figure 6-18: Preference induced by $\mathcal{L}_{X_1} f_1$.

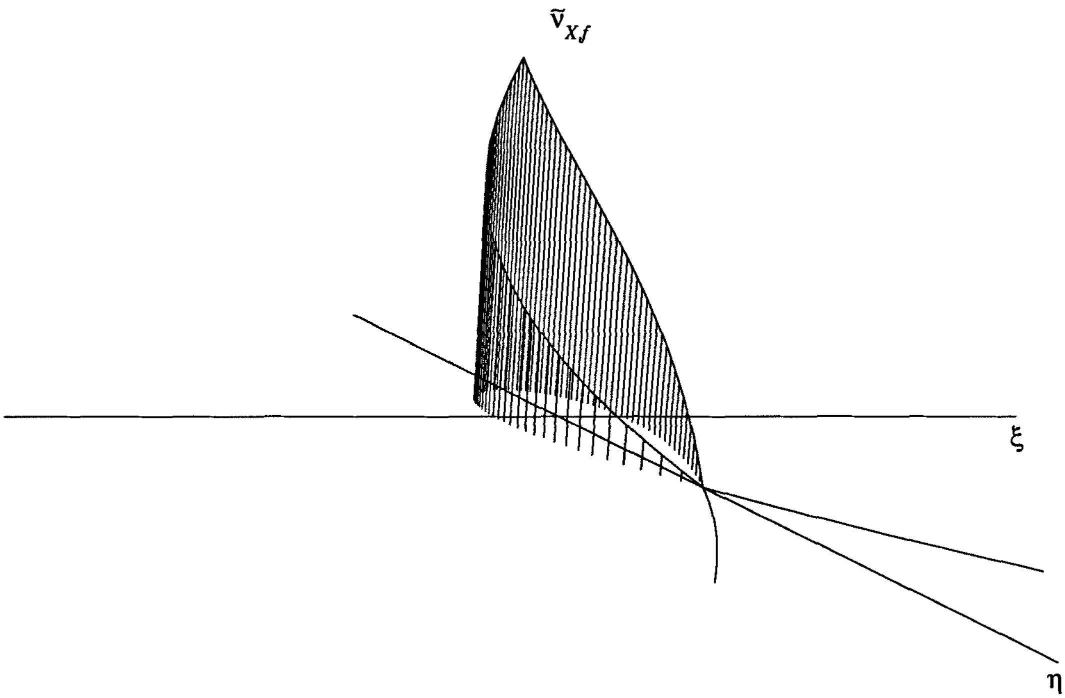


Figure 6-19: Preference induced by f_1 and $\mathcal{L}_{X_1} f_1$.

computed by Equation 6.19 given the preference μ on M. Observe that the projection of the preference curve onto the $(\eta, \tilde{\nu}_{X,f})$-plane consists of a single point. This reflects the fact that f_2 is constant along X_2 (*i.e.*, $\mathcal{L}_{X_2} f_2 = 0$).

Example: Let M, μ, and X_2 be the same as above. Define a function f_3 on M by $f_3(x, y, z) = y^2 + z^2$. The preference on $\mathbb{R}^2$ as computed by Equation 6.19 is shown in Figure 6-22. A straightforward calculation gives $\mathcal{L}_{X_2} f_3 = -2xy$. Observe that f_3, unlike f_2, is not constant along integral curves of X_2. This is verified by the fact that the projection of the preference curve in Figure 6-22 onto the $(\eta, \tilde{\nu}_{X,f})$-plane is nontrivial. Also, the projection of the preference curve onto the $(\xi, \tilde{\nu}_{X,f})$-plane yields a triangular preference function increasing linearly from 0 to 1. This reflects the fact that f_3 has been chosen to be the same as the preference μ.

Preference Induced by Flows of Vector Fields

In this section, ordinary differential equations (o.d.e.'s) will be presented with one, or both, of the following effects of imprecision.

1. The modification of the preference on the variables as they evolve according to the flow of the o.d.e., and
2. the existence of imprecision in parameters in the o.d.e.

A means of computing preferences induced by the flow in these instances is presented. Cases 1) and 2) are examples of a more general problem that is formulated first. In keeping with the spirit of the previous discussion, o.d.e.'s are formulated in the language of vector fields on a manifold.

As in the previous sections, M will be a smooth manifold. Since vector fields may depend on parameters, another manifold D is needed which will be the *parameter space*. Thus now the DPS can be thought of as $M \times D$, but the o.d.e's are over M. Let $\pi_M : M \times D \to M$ be projection onto the first factor. The *tangent bundle* of M will be denoted by TM and $\tau_M : TM \to M$ will denote the tangent bundle projection. As in Lang [89], the *pull-back bundle* $\pi_{MD} : \pi_M^* TM \to M \times D$ can be formed, where

$$\pi_M^* TM = \{(v, (m, \mathfrak{d})) \in TM \times (M \times D) \mid \tau_M(v) = \pi_M(m, \mathfrak{d})\} \tag{6.20}$$

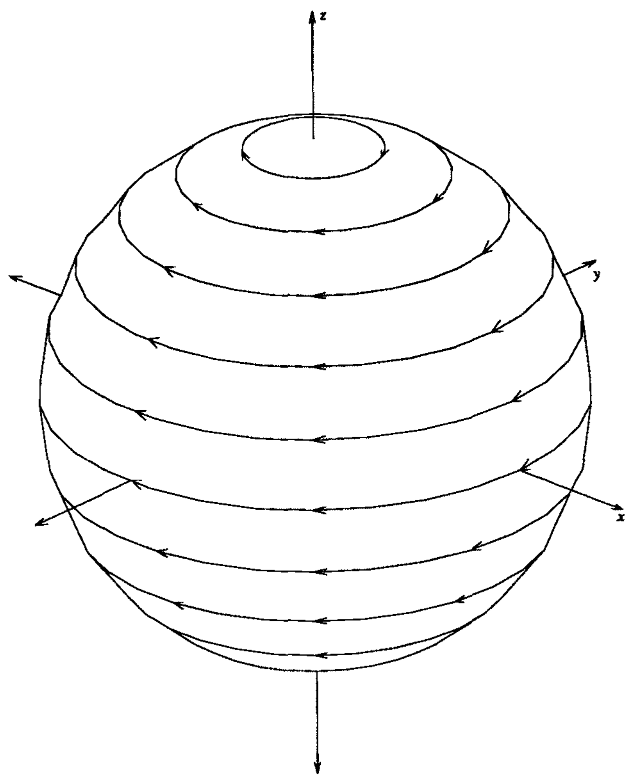


Figure 6-20: Integral curves for X_2 on S^2.

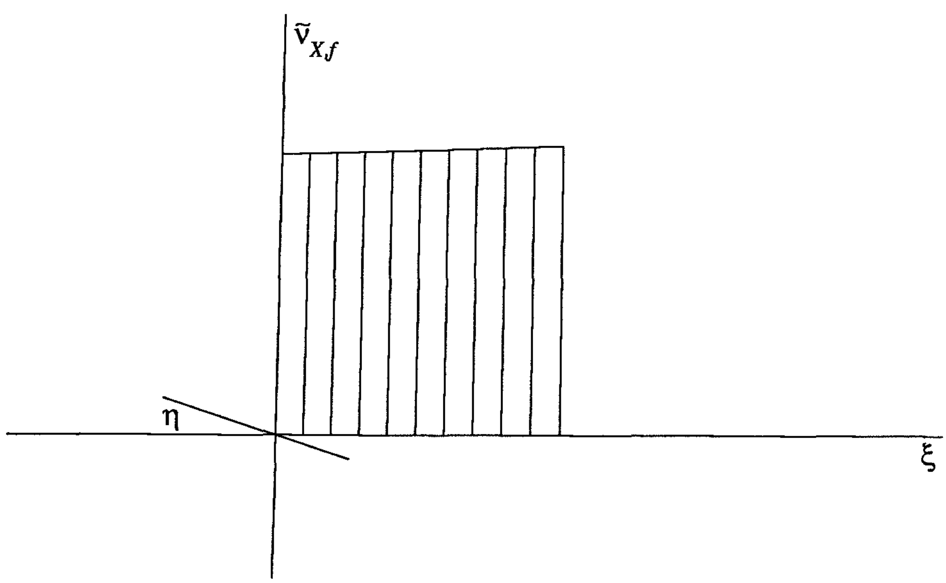


Figure 6-21: Preference induced by f_2 and $\mathcal{L}_{X_2} f_2$.

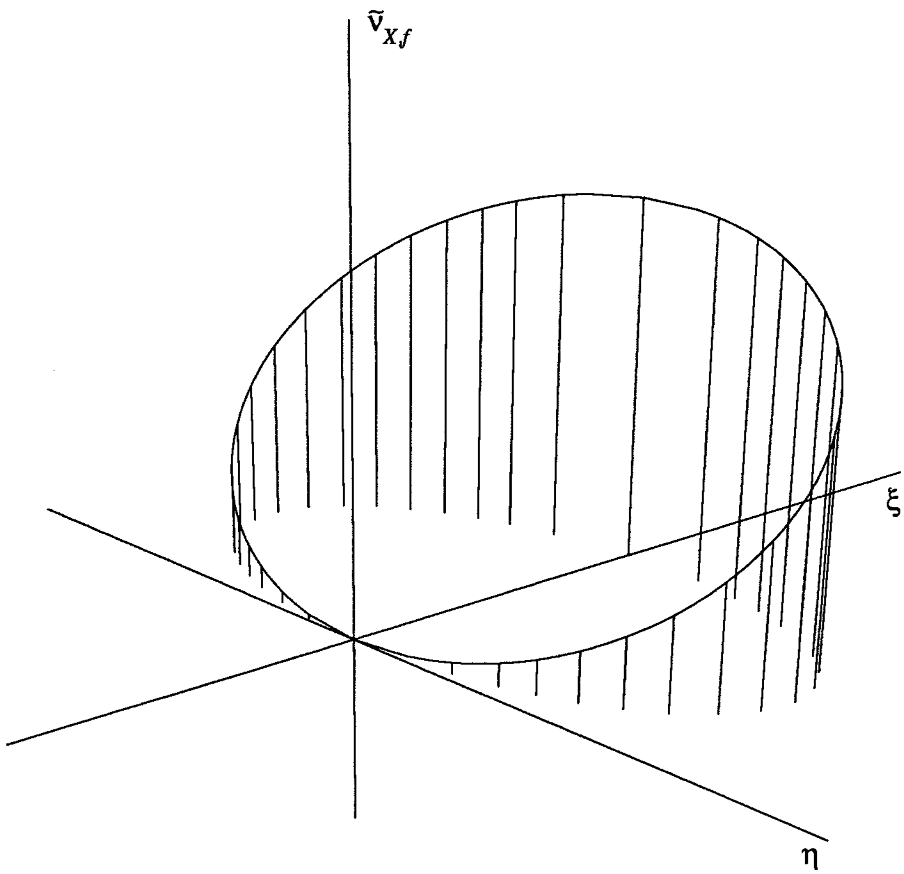


Figure 6-22: Preference induced by f_3 and $\mathcal{L}_{X_2} f_3$.

and

$$\pi_{MD}(v,(m,\mathfrak{d})) = (m,\mathfrak{d})$$

The manifold $\pi_M^* TM$ is to be regarded as a vector bundle over $M \times D$. The fiber over a point $(m,\mathfrak{d}) \in M \times D$ is $\pi_{MD}^{-1}(m,\mathfrak{d}) = T_m M$. Now one can define vector fields depending on parameters.

Definition 6.7 *A* vector field with parameters *on M is a smooth section $X : M \times D \to \pi_M^* TM$ of $\pi_M^* TM$ such that for every $\mathfrak{d} \in D$, $X_\mathfrak{d} : M \to TM : m \mapsto X(m,\mathfrak{d})$ is a vector field on M.*

If $X_\mathfrak{d}$ is the vector field on M corresponding to the parameter $\mathfrak{d} \in D$, its *flow* will be the one-parameter family of diffeomorphisms of M which will be denoted by $F_t^\mathfrak{d} : M \to M$. All vector fields considered here will be complete.

Now suppose a preference, μ, is given on DPS $= M \times D$. For each $t \in \mathbb{R}$, a preference induced on M (Definition 6.3) by a vector field with parameters can be defined as follows.

Definition 6.8

$$\begin{aligned} \nu_t \ &: \ M \to [0,1] \\ &\quad m \mapsto \sup\{\mu(m',\mathfrak{d}) \mid (m',\mathfrak{d}) \in M \times D, \quad F_t^\mathfrak{d}(m') = m\} \end{aligned}$$

If $\bigcup_{\mathfrak{d}\in D} F_{-t}^\mathfrak{d}(m) = \emptyset$, then take $\nu_t(m) = 0$.

The cases 1) and 2) above can be thought of as special cases of Definition 6.8 as follows.

1. *Crisp parameters:* In this case, an initial preference function, μ_M, is given on M. Since there is no dependence on parameters, D is taken to be the manifold consisting of a single point $\mathfrak{d}$ so that what results is the single vector field $X_\mathfrak{d}$ on M with flow $F_t^\mathfrak{d}$. Now define a preference μ on $M \times D$ by $\mu(m,\mathfrak{d}) = \mu_M(m)$. Applying Definition 6.8 will determine how the preference μ_M will evolve under the flow of $X_\mathfrak{d}$. See the first example below.

2. *Imprecise parameters:* In this case, a point $m_0 \in M$ is fixed and a preference on D gives rise to an induced preference on M as m_0 is mapped under the flow of the vector field for the various parameter values. Suppose μ_D is a given preference on D. A

preference can be defined on $M \times D$ by $\mu(m, \mathfrak{d}) = \mu_D(\mathfrak{d})$ if $m = m_0$ and $\mu(m, \mathfrak{d}) = 0$ otherwise. Computing the induced preference given by Definition 6.8 will give the desired preference on M for each t. See the second example below.

Example: In this example, the first case is illustrated. Let $M = \mathbb{R}^2$ and let (x, y) be the standard coordinates for M. Define a vector field $X_4 = -x\partial/\partial x + y\partial/\partial y$ on M. The flow of this vector field is given by the one-parameter family of diffeomorphisms $F_t^4 : M \to M : (x, y) \mapsto (xe^{-t}, ye^t)$. Figure 6-23 shows how two particular preference functions on M starting at t_0 and t_0' are mapped by the flow to preferences on M at times t_1, t_2, and t_1', t_2', respectively. This is a simple example. More complicated flows will give rise to more distorted induced preferences.

Example: In this example, the second case is illustrated. Let $M = \mathbb{R}$ with x the standard coordinate on M. As a vector field on M take $X_5 = ax\partial/\partial x$, where $a \in \mathbb{R}$ is a parameter and has the same preference function as was used for x previously, whose graph is depicted in Figure 6-8. The flow of X_5 is given by $F_t^5 : \mathbb{R} \to \mathbb{R} : x \mapsto xe^{at}$. In Figure 6-24 the preference μ_t on M is computed for two times, t_1 and t_2. The initial point $x_0 \in \mathbb{R}$ has been chosen to be positive. Observe that at $t = 0$, the induced preference on M has support $\{x_0\}$.

Related Work

Others have considered calculus with fuzzy quantities, related to the development here. Negoiţă and Ralescu [102] discuss a similar notion in the context of iterating a set map $f : X \to X$ that depends on imprecise parameters. Novàk [105] extends these ideas to include cases where imprecise initial conditions are mapped under the set mapping f. This situation is included in Definition 6.8.

Dubois and Prade [44, 45, 46] consider induced preference under differentiation on $\mathbb{R}$, as does Goetschel and Voxman [61]. Puri and Ralescu [127] consider differentials on Banach spaces when μ represents a fuzzy number. Kaleva [82] and Seikkala [152] consider o.d.e.'s with fuzzy numbers on $\mathbb{R}^n$ and $\mathbb{R}$, respectively. This section generalizes these works.

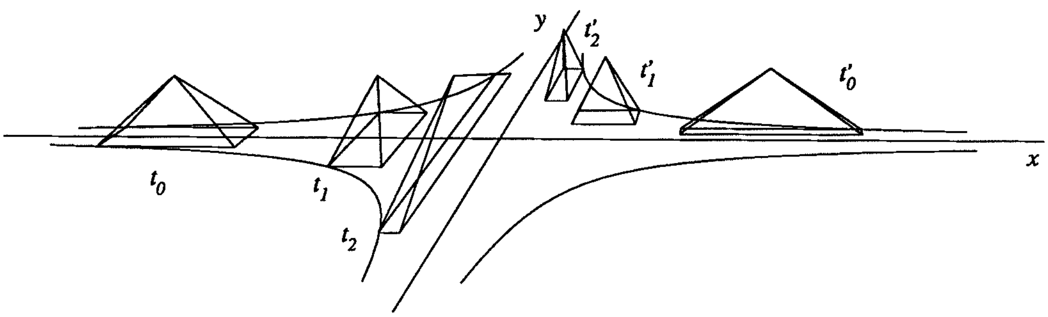


Figure 6-23: Preference induced by the flow of X_4.

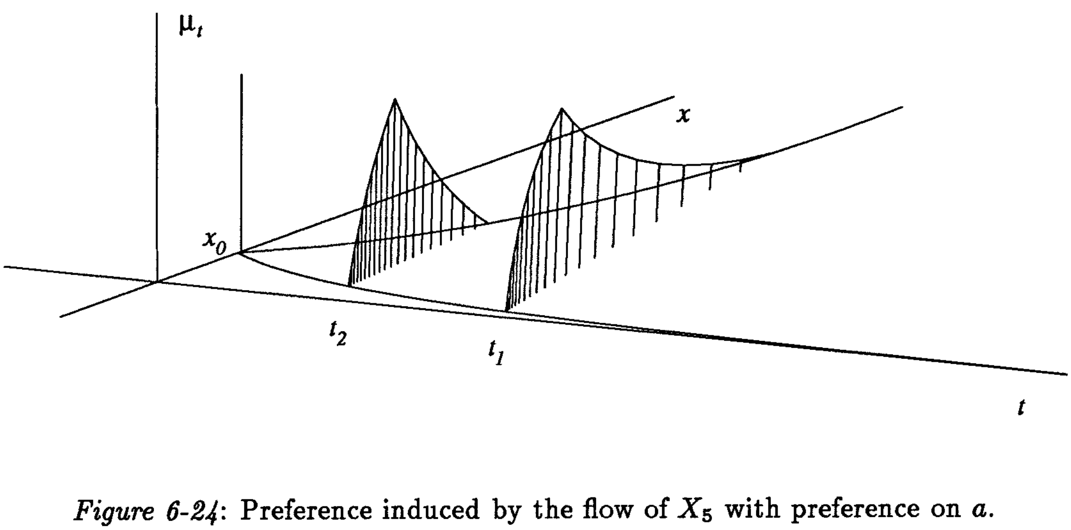


Figure 6-24: Preference induced by the flow of X_5 with preference on a.

Discussion

Methods have been presented for propagating preference on manifolds through the operation of Lie differentiation, and by the flow of vector fields that may or may not depend on imprecise parameters.

Interpretations for preferences induced by differentiation are shown. It is demonstrated that one must consider the total performance space of the function and its derivative simultaneously, to correctly interpret the preferences (Equation 6.19). The induced preferences cannot be considered independently, as with independent functions.

Interpretations for preferences induced by the flows of vector fields are readily made. Figures 6-23 and 6-24 agree with what is expected intuitively. Namely, in Figure 6-23 the initial preference on $\mathbb{R}^2$ is simply transported along the integral curves of the vector field, and in Figure 6-24, an initial preference concentrated at the point x_0 is spread out over $\mathbb{R}$ as time elapses. Even for examples of this type, however, some rather complicated behavior can emerge. For example, if the chosen range of parameters includes a *bifurcation value*, the qualitative behavior of the induced preference could change drastically depending on the parameter.

This section closes the discussion of examples of induced preference calculation. Consequences and claims on induced preference will now be explored.

6.3 Induced Preference Invariance

The previous section defined and discussed induced preference. To perform the calculation of induced preference ν, preference specifications μ must be made, a set map f must be known, and a strategy $\mathcal{P}$ must be made for combining any independent μ. Specification of a strategy presents difficulty in the preliminary design phase (and thus must be iterated over to select). Perhaps, however, there are problems for which the resulting induced preference ν is invariant of the strategy selected. Such problems would present a clear simplification in problem formulation.

As a motivational example, consider again Example 4, the structural truss example. Again, the truss is intended to support a weight W a distance l from a wall, as shown in

Figure 6-2, and discussed in Section 6.2.2. The width and thickness of the beam is w and t respectively, forming the DPS $\simeq \mathbb{R}^2$, with imprecise specifications, as graphed in Figure 6-3. In this example, a typical performance parameter considered was the maximum bending stress in the horizontal bar. This is given by Equation 6.7. The designer must ensure that the bending stress is not excessive; as such, the designer wishes to induce the preferences specified on the DPS to the PPS, to see what resulting stress occurs at the various preference levels. So given these specifications, the designer performs the imprecision transformation calculations, as reflected by Definition 6.4 (with DPS $= \mathbb{R}^2$, and PPS $= \mathbb{R}$).

To do this, a preference for a point $(w, t) \in$ DPS must be defined, given the individual preferences specifications. A strategy must be set, as discussed in Section 6.2.2, where a non-compensating strategy was considered. Now two different cases will be considered. The first is again the non-compensating combination of $\mu_1(w,t) = \min\{\mu_w(w), \mu_t(t)\}$. Also, $\mu_2(w,t) = (\mu_w(w) \cdot \mu_t(t))^{\frac{1}{2}}$ will be considered.

Given these definitions, the induced preference ν is calculated using the imprecision transformation, Definition 6.4, for both combination metrics. This example is significant in that both combination functions μ_1 and μ_2 result in the exact same induced preference ν through σ, as calculated using the imprecision transformation, Definition 6.4, (graphed in Figure 6-4).

Thus, this feature allows problem formulation simplification, in that a designer need not worry about what strategy to use when calculating the imprecision transformation, the results are invariant. Further, the algorithms of Section 6.2.2 could also then be applied to combination metrics other than *min*. The conditions under which this is true have been explored, and the results are summarized in this section. The details and proofs are presented in Appendix B.

The most general means of combining the preferences $\{\mu_i\}$ to now be explored (in this section) is the following:

Definition 6.9 *Let μ_i be a preference on the i^{th} coordinate in $\mathbb{R}^n$. A* combination function *is a continuous function $\mathcal{P} : [0,1]^n \to [0,1]$ so that the function $\mu(d^1, \ldots, d^n) = \mathcal{P}(\mu_1(d^1), \ldots, \mu_n(d^n))$ on $\mathbb{R}^n$ has its support contained in $D_0^1 \times \cdots \times D_0^n$.*

In this section, μ is a preference function on $\mathbb{R}^n$ that is zero at least whenever $\min\{\mu_1(d^1),$

$\ldots,\mu_n(d^n)\}$ is zero. Whenever $\mu(d)$ is written, it is regarded as a preference function arising from a combination function. Definition 6.9 is a more general preference metric than developed in Chapter 4 (the set *IMC*, Definition 4.5): the only restrictions here are that $\mathcal{P}$ be continuous and that the boundary conditions of *min* be used. Thus, $\mathcal{P}$ from Definition 6.9 is a relaxation from *IMC* in idempotency and monotonicity. This is all that is required of the proofs.

The α-cuts for the preference μ will again be denoted by D_α. With this preference on $\mathbb{R}^n$, the induced preference ν on $\mathbb{R}$ for any function $f : \mathbb{R}^n \to \mathbb{R}$ is given by Definition 6.4. The α-cuts of ν will be denoted by W_α. Thus $W_\alpha \subset \mathbb{R}$.

As well, consider the results of using the *min* combination metric. Let P_α denote the α-cut of ν when using the *min*. Then, if f is continuous monotonic, it is known that the LIA algorithm can be applied. Now, the algorithm is blindly applied to any continuous f.

For this discussion, define:

Definition 6.10 *Let* $\alpha \in [0,1]$ *and let* $(d^1,\ldots,d^n) \in D^1_\alpha \times \cdots \times D^n_\alpha$. $(d^1,\ldots,d^n)$ *is said to be an* α-ridge point *if* $(d^1,\ldots,d^n) = (d^1_{\beta_1}(\alpha),\ldots,d^n_{\beta_n}(\alpha))$ *where* $\beta_i \in \{min, max\}$ *for* $i = 1,\ldots,n$.

The set of all α-ridge points will be denoted by $\mathcal{R}_n(\alpha)$. This implies $|\mathcal{R}_n(\alpha)| = 2^n$. These are the points used in the LIA algorithm (Algorithm 6.1). It is desired to use this finite set to compute a closed interval $P_\alpha \subset \mathbb{R}$ that will be the proposed α-cut for the preference on $\mathbb{R}$ induced by f. This also defines the values

$$\begin{aligned} p^\alpha_{min} &= \min\{f(d) \mid d \in \mathcal{R}_n(\alpha)\} \\ p^\alpha_{max} &= \max\{f(d) \mid d \in \mathcal{R}_n(\alpha)\} \end{aligned} \tag{6.21}$$

in $\mathbb{R}$ (the end points of P_α in Algorithm 6.1), and the sets

$$\begin{aligned} P^\alpha_{min} &= \{p \in \mathbb{R} \mid p < p^\alpha_{min}\} \\ P^\alpha_{max} &= \{p \in \mathbb{R} \mid p > p^\alpha_{max}\} \end{aligned} \tag{6.22}$$

This implies $P^\alpha_{\min} \cup P_\alpha \cup P^\alpha_{\max} = \mathbb{R}$. Now the geometric hypotheses on the level sets of the function f and the combination function $\mathcal{P}$ will be presented.

Definition 6.11 *Let $\mathcal{P}$ be a combination function with corresponding preference function μ. Let D_α denote the α-cut of μ. Let $f : \mathbb{R}^n \to \mathbb{R}$ be any function. The pair $(\mathcal{P}, f)$ is said to be* strongly inclusive *if:*

$$\textbf{H1:} \quad D_\beta \cap f^{-1}(P^\alpha_{min}) = \emptyset \quad \textit{for all} \quad \beta \geq \alpha$$

and

$$\textbf{H2:} \quad D_\beta \cap f^{-1}(P^\alpha_{max}) = \emptyset \quad \textit{for all} \quad \beta \geq \alpha$$

and if f is continuous.

The name *strongly inclusive* is motivated by the following proposition:

Proposition 6.1 *Let $(\mathcal{P}, f)$ be strongly inclusive, and let μ be the preference corresponding to $\mathcal{P}$. The preference induced on $\mathbb{R}$ by f is denoted by ν. The α-cut for ν is denoted by W_α. Then $W_\alpha \subseteq P_\alpha$.*

Therefore, under the hypotheses **H1** and **H2** and continuity of f, the LIA α-cuts P_α provide an upper bound for the actual α-cuts W_α. The name *strongly inclusive* comes from the fact that the opposite inclusion, $P_\alpha \subseteq W_\alpha$ holds under much weaker hypotheses on $\mathcal{P}$ and f. This motivates the following definition.

Definition 6.12 *Let $\mathcal{P}$ be a combination function, and let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. The pair $(\mathcal{P}, f)$ is said to be* weakly inclusive *if $\mathcal{P} \in IMC([0,1]^n)$ and if f is continuous.*

Actually, as detailed in Appendix B, only the boundary conditions, monotonicity, and idempotency restrictions of *IMC* is required.

Proposition 6.2 *Let $(\mathcal{P}, f)$ be weakly inclusive, and let W_α be the α-cut for the preference induced on $\mathbb{R}$ by f. Then $P_\alpha \subseteq W_\alpha$.*

Propositions 6.1 and 6.2 determine upper and lower bounds, respectively, for the α-cuts W_α given various hypotheses on the combination function $\mathcal{P}$ and the function f. Now conditions will be stated that must be satisfied if the LIA induced α-cuts P_α provide an exact description of the actual α-cuts W_α. This is important for computational purposes since it immediately distinguishes cases when the LIA algorithm *cannot* be used.

Proposition 6.3 *Let $\mathcal{P}$ be a combination function and let $f : \mathbf{R}^n \rightarrow \mathbf{R}$ be a function. If W_α is the α-cut for the preference induced on $\mathbf{R}$ by f, and if $W_\alpha = P_\alpha$ then the pair $(\mathcal{P}, f)$ satisfies* **H1** *and* **H2**.

The next theorem follows immediately from Propositions 6.1 and 6.2.

Theorem 6.4 *If $(\mathcal{P}, f)$ is weakly inclusive (i.e., $\mathcal{P} \in IMC([0,1])$ and f is continuous), then $W_\alpha = P_\alpha \iff (\mathcal{P}, f)$ is strongly inclusive (i.e.,* **H1** *and* **H2** *are satisfied by $(\mathcal{P}, f)$).*

Observe that if two different pairs $(\mathcal{P}_1, f)$ and $(\mathcal{P}_2, f)$ are both weakly and strongly inclusive with the same f, then the resulting ν does not depend on the choice of $\mathcal{P}_1$ or $\mathcal{P}_2$. Thus identical induced preferences can arise from different combination functions. As can be verified, this is the situation arising in the structural truss example, Example 4, discussed at the start of this section.

6.4 Backwards Path

The last section presented results on induced preference invariance. A further question is the relation between induced preference ν and the development of Chapter 4. Chapter 4 defined the most preferred design parameter values when there is a complete specification of preference: μ is defined over all parameters. This section will now explore the relation between the *forward calculation* of induced preference ν, and the determination of the overall most preferred design parameters.

Suppose now the designer has performed analysis as described by the method of imprecision, and is prepared to make a decision over the final design parameter values to use, as discussed in Chapter 4. That is, the designer wishes to combine all the preferences $\mu(d)$ on the DPS and $\mu(p)$ from the PPS (via $\mu \circ f(d)$) into an overall preference $\mu_c(d) = \mathcal{P}(\mu(d), \mu \circ f(d))$ on the DPS. This is as done in Chapter 4, with the overall preference denoted μ_c for clarity.

This section now demonstrates that points in the DPS which maximize overall preference can also be determined by another method. One can induce the preferences specified on the DPS $\mu(d)$ onto the PPS by the imprecision transformation, Definition 6.3, to create $\nu(p)$, in a *forward* manner. There one can find the $p \in$ PPS that maximize an overall preference

$\mu_c(p) = \mathcal{P}(\nu(p), \mu(p))$, and then back map to the DPS simply by looking up the values of d used in the original forward mapping ν at the optimal p. This is true even though the induced preference ν on the PPS involves only the preference $\mu(d)$, and does not consider any of the dependent set preferences $\mu(p)$. The results, however, are the points which maximize the overall preference μ_c : the supremum of the combination of the preferences specified both on DPS and PPS. Since this calculation reverses the direction of usage of the induced preference, it is termed the *backwards path* [188].

6.4.1 Definitions and Notations

It is desired to combine preferences on DPS and PPS in the following manner. Let $\mathcal{P} : [0,1]^2 \to [0,1]$ be monotonic in its first argument. Thus, if $0 \leq a \leq b \leq 1$, then $\mathcal{P}(a,t) \leq \mathcal{P}(b,t)$ for all $t \in [0,1]$. In this manner one can make the definitions

$$\mu_{cd}(d) = \mathcal{P}(\mu(d), \mu \circ f(d)) \tag{6.23}$$

and

$$\mu_{cp}(p) = \mathcal{P}(\nu(p), \mu(p)) \tag{6.24}$$

as overall preferences on DPS and PPS, respectively. Thus μ represents specified preference on the parameters, ν represents induced preference on the performance parameters, and μ_c represents calculated overall preference. The objective is to determine the most preferred point in the DPS; that is, the point in DPS that maximizes the overall preference μ_{cd}; μ_{cd} is as developed in Chapter 4.

Since μ_{cd} and μ_{cp} are preference functions, and therefore bounded, they have finite supremum over DPS and PPS, respectively. So the following numbers in $[0,1]$ can be defined.

$$||\mu_{cd}||_\infty = \sup\{\mu_{cd}(d) \mid d \in \text{DPS}\} \tag{6.25}$$

and

$$||\mu_{cp}||_\infty = \sup\{\mu_{cp}(p) \mid p \in \text{PPS}\} \tag{6.26}$$

Further the sets

$$\mathcal{D}^* = \{d \in \text{DPS} \mid \mu_{cd}(d) = ||\mu_{cd}||_\infty\} \tag{6.27}$$

and

$$\mathcal{P}^* = \{p \in \text{PPS} \mid \mu_{cp}(p) = ||\mu_{cp}||_\infty\} \tag{6.28}$$

are assumed not empty. This is true, for example, when DPS and PPS are topological spaces, and μ_{cd} and μ_{cp} are continuous with compact support. It may be true in other cases, however, and these will not be excluded from consideration.

6.4.2 Backwards Path Results

The first result stated will be the foundation of the results to follow.

Proposition 6.5 $\mu_{cp}(p) = \sup\{\mu_{cd}(d) \mid d \in \text{DPS}, f(d) = p\}$

Proof. First, a technical lemma will be proven.

Lemma 6.6 *Let* $\mathcal{Q} : \mathbb{R} \to \mathbb{R}$ *be monotonic, and let* $g : \mathbb{R} \to \mathbb{R}$ *be any function. Then*

$$\mathcal{Q}(\sup\{g(x) \mid x \in \mathbb{R}\}) = \sup\{\mathcal{Q}(g(x)) \mid x \in \mathbb{R}\}$$

Proof. Since $\mathcal{Q}$ is monotonic, the expression on the right-hand side of the equality will be evaluated at the value of x, possibly $\pm\infty$, where g attains its supremum. But this is precisely what the left-hand side of the equation returns. ▼

Now the proof of the proposition will be given. Using Lemma 6.6,

$$\begin{aligned}
\mu_{cp}(p) &= \mathcal{P}(\nu(p), \mu(p)) \\
&= \mathcal{P}(\sup\{\mu(d) \mid d \in \text{DPS}, f(d) = p\}, \mu(p)) \\
&= \sup\{\mathcal{P}(\mu(d), \mu(p)) \mid d \in \text{DPS}, f(d) = p\} \\
&= \sup\{\mathcal{P}(\mu(d), \mu \circ f(d)) \mid d \in \text{DPS}, f(d) = p\} \\
&= \sup\{\mu_{cd}(d) \mid d \in \text{DPS}, f(d) = p\}
\end{aligned}$$

■

The following lemma will be useful in the sequel.

Lemma 6.7 $||\mu_{cd}||_\infty = ||\mu_{cp}||_\infty$

Proof. Choose $p \in \mathcal{P}^*$ so that $\mu_{cp}(p) = ||\mu_{cp}||_\infty$. But $\mu_{cp}(p) = \sup\{\mu_{cd}(d) \mid d \in \text{DPS}, f(d) = p\} \leq ||\mu_{cd}||_\infty$. Thus $||\mu_{cp}||_\infty \leq ||\mu_{cd}||_\infty$. Now let $d \in \mathcal{D}^*$ so that $\mu_{cd}(d) = ||\mu_{cd}||_\infty$. Then $\mu_{cp}(f(d)) = \sup\{\mu_{cd}(d') \mid f(d') = f(d)\} \geq ||\mu_{cd}||_\infty$. Thus $||\mu_{cp}||_\infty \geq ||\mu_{cd}||_\infty$ so $||\mu_{cp}||_\infty$ must equal $||\mu_{cd}||_\infty$. ■

The interest is in ascertaining the peak preference values in DPS given those in PPS.

Theorem 6.8 $f(\mathcal{D}^*) \subset \mathcal{P}^*$. *Conversely, if the DPS and the PPS are topological spaces, and if*

1. *f is continuous, and has compact support, and*

2. *if μ_{cd} is continuous when restricted to $f^{-1}(p)$ for each $p \in \mathcal{P}^*$*

then $\mathcal{P}^* \subset f(\mathcal{D}^*)$.

Proof. Let $d \in \mathcal{D}^*$. Then $\mu_{cd}(d) = ||\mu_{cd}||_\infty$. Then $\mu_{cp}(f(d)) = \sup\{\mu_{cd}(d') \mid f(d') = f(d)\} \geq ||\mu_{cd}||_\infty$. By Lemma 6.7 $\mu_{cp}(f(d)) = ||\mu_{cd}||_\infty$ and so $f(d) \in \mathcal{P}^*$.
Now suppose *i)* and *ii)* hold, and let $p \in \mathcal{P}^*$ so that $\mu_{cp}(p) = ||\mu_{cp}||_\infty$. Since f is continuous, $f^{-1}(p)$ is closed for each $p \in \mathcal{P}^*$ (since such a p is itself closed). Thus, since f has compact support, $f^{-1}(p)$ is also compact (a closed subset of a compact set is also compact). Now μ_{cd} is continuous on the compact set $f^{-1}(d)$ and so assumes its maximum value in the set. But, by definition, $\sup\{\mu_{cd}(d) \mid d \in f^{-1}(p)\} = ||\mu_{cp}||_\infty$, so there is some $d \in f^{-1}(p)$ such that $\mu_{cd}(d) = ||\mu_{cp}||_\infty = ||\mu_{cd}||_\infty$, so $d \in \mathcal{D}^*$. Thus $\mathcal{P}^* \subset f(\mathcal{D}^*)$. ■

In practice, if f is continuous, it may be possible to satisfy the hypothesis of compact support simply by restriction to the domain of interest.

6.4.3 Example 4: The Backwards Path

Consider again the truss example, Example 4. Again, the truss is intended to support a weight W a distance l from a wall, as shown in Figure 6-2. The width and thickness of the beam is w and t respectively, forming the DPS.

Again, given a configuration (as shown in Figure 6-2), typically a designer performs calculations to rate different values of the design parameters. For example, the maximum bending stress in the horizontal bar, given by Equation 6.7, might be calculated, since the designer must ensure that the bending stress is not excessive. As such, now an imprecise specification on the dependent performance parameter space $\mathbb{R}$ is made, as graphed in Figure 6-25.

Now given these specifications, the designer can perform calculations, such as computing the preference induced on the performance parameter space by the map σ and the preference specifications made on w and t, as discussed in Section 6.2. This will be calculated using the imprecision transformation, as given by Definition 6.4 (with $D = \mathbb{R}^2$, and $P = \mathbb{R}$).

Having performed this calculation, the designer can then compare this induced preference ν with the requirement preference μ. For example, the designer may select the optimal value of σ by combining the two preferences with $\mu_{1,cp}(\sigma) = \min\{\nu(\sigma), \mu(\sigma)\}$, or, perhaps with a combination $\mu_{2,cp}(\sigma) = \nu(\sigma)^{\frac{2}{3}} \cdot \mu(\sigma)^{\frac{1}{3}}$. These reflect two different possible combination functions, but the results of Section 2 apply to both. The results of applying the two methods are shown in Figure 6-26.

Consider now the optimal values of σ as defined by the maximum of values of $\mu_{1,cp}$ and $\mu_{2,cp}$. For $\mu_{1,cp}$, the optimal value is $\sigma_1^* = 0.298$ (GPa). For $\mu_{2,cp}$, the optimal value is $\sigma_2^* = 0.295$ (GPa). These define the most preferred values of performance. The question is what are the optimal points w_i^* and t_i^* that produce these performances σ_i^*, for $i = 1, 2$.

It can be shown that the hypotheses *i)* and *ii)* of Theorem 6.8 hold. Therefore the points used in the imprecision transformation calculation of $\nu(\sigma^*)$ are in fact the points used to compute $\mu_{cp}(\sigma^*)$. That is, when using *min* as a combination, at the optimal value of performance $\sigma_1^* = 0.298$ (GPa), the points $w_1 = 0.07003$ (m) and $t_1 = 0.08834$ (m) were used to calculate $\nu(\sigma_1^*)$. As well, when using the product as a combination, at the optimal value $\sigma_2^* = 0.295$ *(GPa)*, the points $w_2 = 0.07697$ (m) and $t_2 = 0.08471$ (m) were used to calculate $\nu(\sigma_2^*)$. Theorem 6.8 demonstrates that these values of w_i and t_i are the optimal points of overall preference $\mu_{i,cd}$, for $i = 1, 2$. That is, if the designer had as well calculated w_i^* and t_i^* defined by

$$\mu_{1,cd}(w_1^*, t_1^*) = \sup\{\min\{\mu_w(w), \mu_t(t), \mu \circ \sigma(w,t)\} \mid (w,t) \in \mathbb{R}^2\}$$

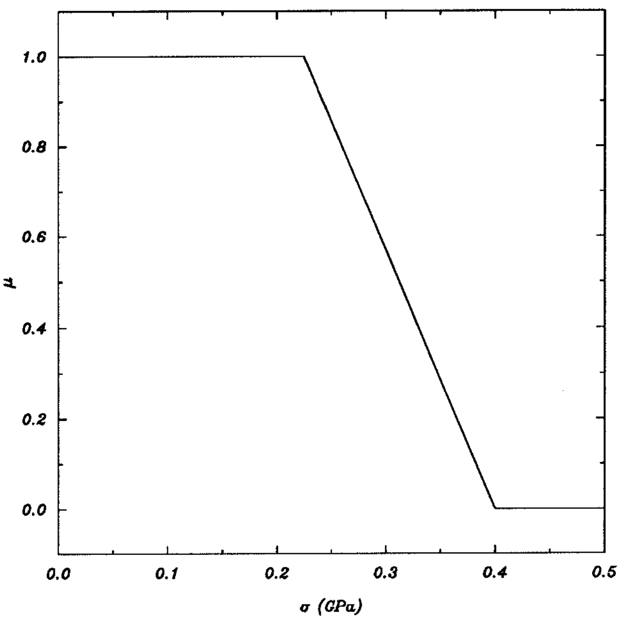


Figure 6-25: Example 4: functional requirement.

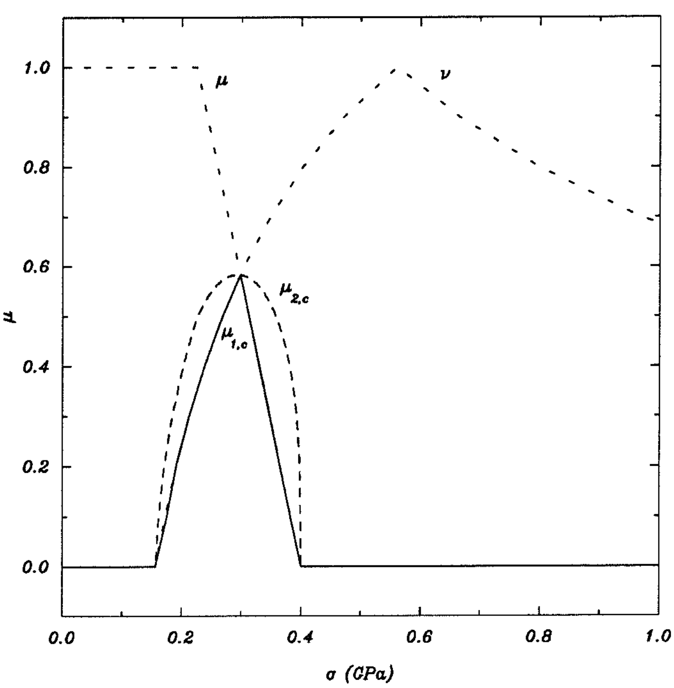


Figure 6-26: Example 4: overall preference on the PPS.

or

$$\mu_{2,cd}(w_2^*, t_2^*) = \sup\{(\mu_w(w) \cdot \mu_t(t) \cdot \mu \circ \sigma(w,t))^{\frac{1}{3}} \mid (w,t) \in \mathbb{R}^2\}$$

respectively (reflecting the two different combination functions ν_{cp} on σ), then the resulting $w_i^* = w_i$ and $t_i^* = t_i$. The key aspect of this is that ν back-maps the optimal points in the DPS of μ_{cp}. Thus one can use the pre-image of the design parameter induced preference ν to find the optimal point of overall preference μ_{cd}, even though ν does not involve the performance parameter preference specification μ.

6.4.4 Discussion

The backwards path was first conceived by Wood and Antonsson [188], where it was developed for a conservative design strategy and single valued preferences on a one-dimensional real PPS. That is, the backwards path was used for the case of PPS $\simeq \mathbb{R}$, and with $\mu(p) = 0$ except for a single value $\overline{p}$, for which $\mu(\overline{p}) = 1$. This section generalizes this work to general functional requirements $\mu(p)$ and a more general performance space PPS (a topological space), and proves the correctness of the backwards path use.

This section presents an observation about preference optimization problems with dependent parameters: one can optimize preference over either the DPS, or over the dependent PPS. One could compose the dependent set preferences onto the DPS and optimize overall preference over the DPS to find the optimal point in the DPS, as in Chapter 4. Alternatively, one could induce the original set preferences onto the PPS and optimize overall preference over the PPS, and then use the pre-image of the induced preference to determine the optimal point in the DPS. This is true although the pre-image is of the induced preference only, not the pre-image of the overall preference.

When using the method of imprecision, one calculates the induced preference of the design parameters on the performance parameter space, to observe the performance achievable in the model. The backwards path is useful, in that it proves one can use the results of the induced preference calculation to perform a different subsequent search more easily: the search for the overall most preferred points.

Thus, the complete *Method of Imprecision* is now presented in Figure 6-27. Again, at the start of the process, the designer suggests possible solutions. These are subsequently

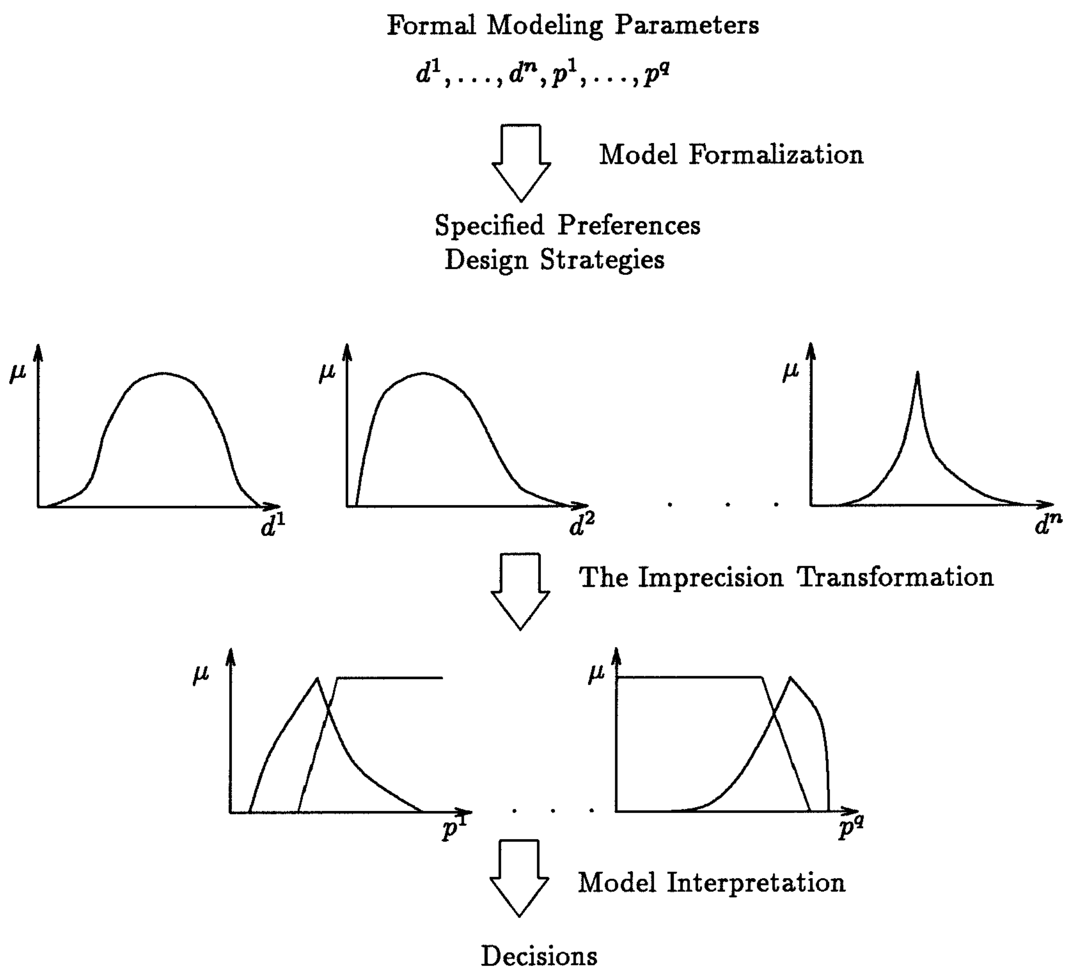


Figure 6-27: The Method of Imprecision.

transformed into formal candidate solution models upon which computations can be performed. Each model includes a DPS and PPS. The designer then makes *a priori* estimates of values for any of the design and performance parameters desired. These preferences are then induced onto the remaining parameters, to observe the effects of any specifications on the remaining parameters.

After having made the induced preference calculations, the designer can observe the imprecise performance achievable and proceed to judge the candidates. Of course, this process can continue even when the designer has preliminary preferences specified on all the parameters. Given any parameter, the induced preference from the other parameters can be calculated, and the ν and μ can be compared. In doing so, any specification μ or $\mathcal{P}$ can be modified, and the resulting ν observed. Finally, when the overall model is accepted, the backwards path can be used to determine the overall most preferred points in the DPS.

This technique is superior to others, since it presents visual information to the designer about the model. An optimization routine will produce a (hopefully globally optimal) solution, but it presents little information about critical aspects in the problem. If the method of imprecision is adopted, it is useful to know that the induced preference pre-image of the optimal performance value is the optimal preference point, since this eliminates the need for a subsequent search.

6.5 Uncontrollable Parameters in Preliminary Design

This chapter has so far assumed that the NPS is trivial. This assumption is now discarded, and a complete NPS as described in Chapter 5 is assumed.

Clearly, the convolution of the development of this chapter and the NPS can be made. The induced preference at a value $u \in \mathrm{U}$ can be calculated from the *expected* pre-image point $s \in \mathrm{S}$ with maximal *expected* preference, with expectation as discussed in Chapter 5.

Thus, nothing in the previous sections of this chapter change, except the evaluations become more difficult, due to the extra calculation required for the expectation process. In any case, the specific equations to be solved are given below for the three parameter forms described. The simpler Definition 6.3 is used instead of Definition 6.1.

For the uncertainty form of probability, the induced preference for any fixed value

$p \in$ PPS can be determined by using the probabilistic expectation. That is, for $d \in$ DPS, determine

$$\overline{\mu}(d) = \int_{\mathrm{NPS}} \mu(d,n)\, dPr \tag{6.29}$$

and

$$\overline{p}(d) = \int_{\mathrm{NPS}} f(d,n)\, dPr \tag{6.30}$$

and then define

$$\overline{\nu}(p) = \sup\{\overline{\mu}(d) \mid p = \overline{p}(d),\ d \in \mathrm{DPS}\} \tag{6.31}$$

or 0 if no such d exist. Thus, $\overline{\nu}$ is the *expected preference* induced from the DPS, given the probabilistic uncertainty.

For the uncertainty form of possibility, the induced preference for any fixed value $p \in$ PPS can be determined by using the possibilistic expectation. That is, for $d \in$ DPS, determine

$$\overline{\mu}(d) = \sup\{\min\{\mu(d,n), \Pi(N)\} \mid n \in N \text{ disjoint} \in \mathcal{B}\} \tag{6.32}$$

and

$$\overline{n}(d) = \{n \in N \text{ disjoint} \in \mathcal{B} \mid \min\{\mu(d,n), \Pi(N)\} = \overline{\mu}(d)\} \tag{6.33}$$

and then define

$$\overline{\nu}(p) = \sup\{\overline{\mu}(d) \mid p = f(d,n),\ n \in \overline{n}(d), d \in \mathrm{DPS}\} \tag{6.34}$$

or 0 if no such d exist. Again, the limit as the partition becomes finer in $\mathcal{B}$ is used. Thus, $\overline{\nu}$ is the *expected preference* induced from the DPS, given the possibilistic uncertainty.

For the uncertainty form of necessity, the induced preference for any fixed value $p \in$ PPS can be determined by using the necessary expectation. That is, for $d \in$ DPS, determine

$$\overline{\mu}(d) = \inf\{\max\{\mu(d,n), \mathfrak{N}_\alpha(N)\} \mid n \in N \text{ disjoint} \in \mathcal{B}\} \tag{6.35}$$

and

$$\overline{n}(d) = \{n \in N \in \mathcal{B} \mid \max\{\mu(d,n), \mathfrak{N}_\alpha(N)\} = \overline{\mu}(d)\} \tag{6.36}$$

and then define

$$\overline{\nu}(p) = \sup\{\overline{\mu}(d) \mid p = f(d,n),\ n \in \overline{n}(d), d \in \text{DPS}\} \tag{6.37}$$

or 0 if no such d exist. Again, the limit as the partition becomes finer in $\mathcal{B}$ is used. Thus, $\overline{\nu}$ is the *expected preference* induced from the DPS, given the necessary uncertainty.

Clearly, these formulations can be convolved for hybrid uncertainty forms, as discussed in Section 5.5. Also, clearly these formulations reduce to the formulation of induced preference with no noise (Definition 6.3) when the NPS is trivial, as occurred in Chapter 5.

6.5.1 Example 3: Preliminary Design of an Air Tank

Consider again Example 3, the air tank example, as discussed in Sections 4.6.2 and 5.6.1. The example is now considered as a preliminary design, and preferences are to be induced from one space to another.

Consider when preferences have been specified on the PPS parameterized by the metal volume m and tank capacity v, as given by Equations 4.16 and 4.17. These preferences can be induced onto the DPS, (parameterized by l and r), by the imprecision transformation. There is uncertainty in the mapping, however, as discussed in Section 5.6.1. Thus, this uncertainty must be included in the mapping, as discussed. This means the preference, induced from the PPS (parameterized by (m, v)) onto the DPS (parameterized by (l, r)), is defined as

$$\begin{aligned} \nu(l,r) \;=\; & \sup\{\inf\{\textstyle\int_S \int_{\delta r} \int_{\delta l} \min\{\mu_m, \mu_v \mid (m,v) = f(l,r,\delta l,\delta r,S,E,P)\} \\ & \times pdf(S)\, pdf(\delta r)\, pdf(\delta l) \times dS\ d(\delta l)\ d(\delta r) \\ & \mid (P,E) \in \mathcal{N}_{0.9}\} \mid (m,v) \in \mathbb{R}^2\} \end{aligned} \tag{6.38}$$

where f is reflected by Equations 4.16 and 4.17. This will present the preference specifications made on the PPS, as graphed in Figure 4-7 and 4-8, as they are expected to appear on the DPS, given the uncertainty space. Also, a non-compensating design strategy is used. The result is graphed in Figure 6-28.

Thus, the designer can simply ignore making preference specifications on some aspects of the tank design, and induce the preferences to observe possible performance. A designer need not induce the preferences onto the design space, however. One could, for example, induce any design parameter preferences onto the PPS. Consider when preferences on the tank design are as graphed in Figure 4-5 and 4-6. These can be induced onto the PPS (parameterized by (m, v)).

$$\begin{aligned} \nu(m,v) &= \sup\{\inf\{\textstyle\int_S \int_{\delta r} \int_{\delta l} \min\{\mu_l, \mu_r \mid (m,v) = f(l,r,\delta l,\delta r,S,E,P)\} \\ &\quad \times pdf(S)\, pdf(\delta r)\, pdf(\delta l) \times dS\; d(\delta l)\; d(\delta r) \\ &\quad \mid (P,E) \in \mathcal{N}_{0.9}\} \mid (l,r) \in \mathrm{R}^2\} \end{aligned} \tag{6.39}$$

where f is reflected by Equations 4.16 and 4.17. This will present the preference specifications made on the DPS, as graphed in Figure 4-7 and 4-8, as they are expected to appear on the PPS, given the uncertainty space. Also, a non-compensating design strategy is used. The result is graphed in Figure 6-29.

6.6 Conclusions

This chapter has presented a formal technique for solving engineering design problems. This chapter is thus a culmination of the preceding work, in that it presents a constructive methodology for solving engineering design problems, using the previous development. This is different from the previous chapters, which illuminated consequences of particular formalizations. The previous chapters defined formal properties; this chapter used them.

A method to perform engineering design calculations under imprecision is described. Interval methods are presented to realize an efficient algorithm for such calculations. Invariance properties of the results were explored, and geometric conditions were found demonstrating when invariant induced preference can occur. Though the conditions are not "clean" algebraic conditions which can be easily identified, this inability is itself a theoretical result.

Some simple examples are also included to illustrate possible anomalies with imprecise design calculations. These simple examples demonstrate the ill-conditioning that can occur. Such formal ill-conditioning was given meaning informally. Examples were also presented on parameter spaces more complex than simply R^n. Definitions were given for calculating with preference on manifolds, and with differential equations. The imprecision transformation is

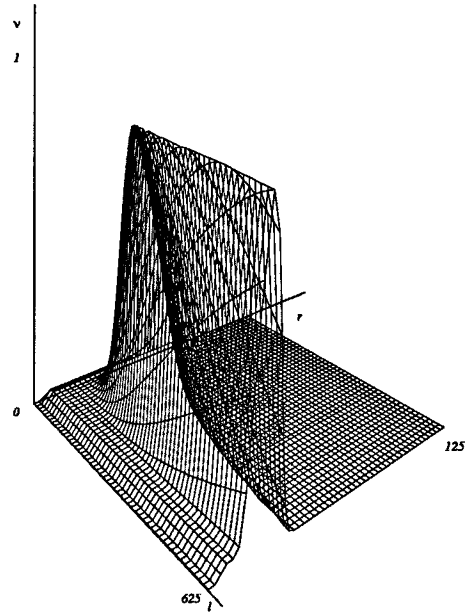


Figure 6-28: Example 3: $\nu_{(m,v)}$ over the DPS.

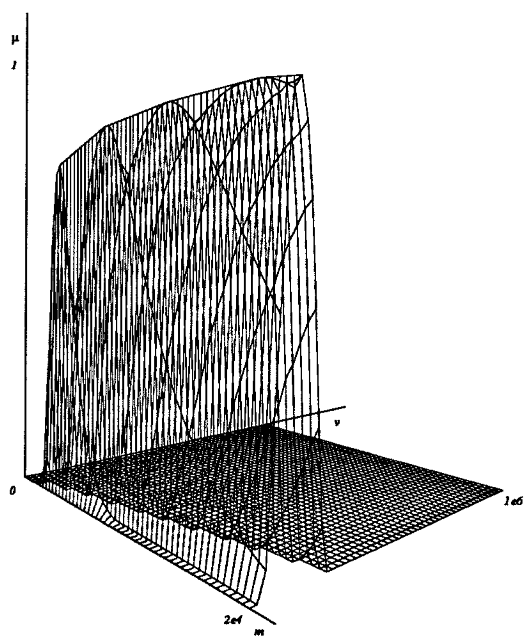


Figure 6-29: Example 3: $\nu_{(l,r)}$ over the PPS.

defined using set mappings; thus, it can be generally applied.

When applying the imprecision transformation, a computational aspect was demonstrated. When preference is induced onto a dependent space, the point used in the inducement to the optimal value is as well the optimal point for the entire problem. This is proven true when the spaces have topological properties, which is quite general. Thus, though the imprecision transformation is computationally difficult, this concern is muted by the fact that a subsequent search (as applied in Chapter 4) is not needed.

This methodology is deemed superior to other methods that instead attempt to initially identify every possible source of designer satisfaction, and then conduct a search over the alternatives, based on maximizing this initial formalization. That is, other methods pose artificial questions to designers about their preferences, and adopt rationality beliefs to extrapolate to "most preferred" solutions. A framework is formed in which the *a priori* preferences are supposed to be accurate. All possible sources of preference specification are assumed to be identified. The solutions to such techniques are known to instead produce results that are dependent on the artificial questions.

This research, on the other hand, adopts a position that one need not attempt to formulate such preferences accurately initially. Rather, this research considers the belief that a designer can pose an *a priori* preference as the designer believes it to be, and then compute with it. When iterating, new understanding arises which can allow the designer to form new beliefs and modify the initial specifications.

This methodology can allow a designer to formally compute on incomplete designs. "What if" questions are easily framed in the method of imprecision. Preliminary preference specifications can be made, and propagated through a model with the imprecision transformation. Different combination metrics can be applied, and the results propagated through a model. Different noise characterizations can be made, and their effects propagated through a model. This can be performed iteratively, converging to a final design.

Chapter 7

Conclusions

Having developed an interpretation of formal engineering design, the work is circumscribed. Formal models have been defined in the context of engineering design. A paradigm for the formal-informal interaction of a design process has been developed.

7.1 Design Formalization

A belief behind this thesis is that one can delineate between informal and formal concepts in engineering design. Precise definitions of "formal" and "computation" are given for the engineering design domain. Formal models have set structure. Computations are mathematical operations on sets. Informal interpretations are generally without set structure: they are thoughts, beliefs, and levels of contentment. Informal reasoning may behave like mathematical operations, or it may not.

Such concepts are important to engineering design, since it incorporates both formal computation and informal interpretation. Design as an activity occurs to satisfy a need: "need" and "satisfy" are informal. Engineering design is defined to require computation. Thus engineering design exhibits both informal and formal concepts.

This characterization provides a framework to analyze current engineering practices. Many references discuss informal methods for generating designs [9, 90, 121]. Others involve both formal and informal concepts. QFD [4, 69] uses a formal matrix to formally rate formal alternatives. Yet the formal alternatives usually represent informal concepts, since the alternatives are not yet fully designed. This interaction that naturally occurs in engineering design has been an object of study in this thesis.

7.1.1 Canonical Activities in an Engineering Design Process

It is clear that there are different activities that occur in engineering design. These activities can be characterized into a nominal conceptualization of how a design process can proceed. A desire must first be recognized. This is a canonical activity of engineering design: it always occurs, though it may be described in different terms with different design processes.

A second canonical activity of engineering design is the formalization process. This is usually considered to be "the" design process [2, 166, 197]: developing an informal idea to the point where one can analyze the design. The result is a formal model: a description on which computation can occur.

A third canonical activity of engineering design is formal computation: to formally make performance evaluations at points in the set of alternatives. Engineering design research has made the most development researching this activity [1].

An argument presented in this work is that a fourth canonical activity of engineering design is designer interaction with a formal model. To some degree, it always occurs. Designers make formal evaluations in engineering design, and informally consider the formal results.

A final canonical activity of engineering design is informal interpretation: customers evaluate the produced design. They evaluate an actual device as manufactured. This distinguishes this activity from the previous in two respects: first, the customer, not the designer evaluates the result. Second, the customer evaluates the device, not a formal model of the device.

This characterization of a design process has occurred many times before [9, 56, 121]. This section simply re-presents it, but making clear the informal and formal aspects of such a characterization.

7.2 Formal Model Interaction

A central theme of this thesis has been the interaction between informal designer interpretations and formal model computations. The interaction between a designer's satisfaction and formal computation was pursued. The conduit for this research was with preference.

The satisfaction of a designer with a design goal was formally modeled on a real scale. Such a methodology is quite general; the text discussed simplifying restrictions to reduce to other techniques. The price paid for generality is in work. The designer must specify preferences, weights, and strategies. That is rather difficult to do. The work behind this thesis, however, did not primarily seek to develop techniques for "doing" design. Rather, the fundamental question pursued was *what* should be solved for in engineering design. Clearly, some *how* questions were answered along the way, as witnessed by the development of Chapter 6. But such development remained in the context of what occurs during any designer interaction with a formal model.

This work demonstrates issues involved when determining the solution to a design problem. These issues *always* exist; they may be determined by simplifying assumptions of some techniques, but nevertheless the issues are there. Thus, this work creates a framework to analyze new and existing techniques. With the development of the method of imprecision, a formal/informal iterative design process of specification, calculation, observation, and re-specification has been given foundation.

Appendix A

Determining Preferences

A.1 A Lottery Method

This section will review a basic *lottery method* for establishing a preference over values of a goal [53, 58, 129]. Consider a finite dimensional goal X, as defined in Chapter 4.

Over X, the designer must establish an ordering, based on satisfaction, such that, for $x_i \in X$:

$$x_1 \preceq \ldots \preceq x_j \preceq \ldots \preceq x_M \tag{A.1}$$

Thus, x_1 satisfies the least in X, and x_M satisfies the most in X. Then, for all j, the designer must be asked the *lottery question*:

On a scale of zero to 1, what is your belief $\mu(x_j)$ that you are indifferent between:

1. *The certain option: receive x_j.*
2. *The risky option: receive x_M with certainty $\mu(x_j)$, and receive x_1 with certainty $1 - \mu(x_j)$?*

This constructs a dependent set $\{\mu(x_j)\}$ with a real ordering

$$\mu(x_1) \leq \ldots \leq \mu(x_M) \tag{A.2}$$

which forms a measurement of preference.

If the space X is uncountable, such questions can be asked on the countably dense subset.

A.2 Methods for Determining Weights

Consider a point $d \in$ DPS, with an evaluated performance $p \in$ PPS, $p = f(d)$. Then preferences can be established on each x_j as developed in Chapter 4,

$$\mu(x_1), \ldots, \mu(x_N) \tag{A.3}$$

at each point $d \in$ DPS. If weights are to be used, it remains to determine them at each point $d \in$ DPS.

A.2.1 Analytic Hierarchy Process

One method available for eliciting weighting functions is the *Analytic Hierarchy Process*, conceived by Saaty [141, 142]. Using this method, one can ask the designer the comparison question, $\forall\, i, j \in \{1, \ldots, N\}$:

> *On a scale of one to ten, how much more important is goal x_i than x_j, given the preferences $\mu(x_i)$, $\mu(x_j)$?*

This constructs a set of relative, pair-wise comparisons $\{c_{ij}\}$. If the designer is consistent between the x_i, x_j and x_j, x_i comparison, then only $\frac{N(N-1)}{2}$ comparison questions need be asked.

The comparisons can be placed in matrix of the form

$$W = \begin{bmatrix} 1 & c_{21} & \ldots & c_{N1} \\ \frac{1}{c_{21}} & 1 & \ldots & c_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{c_{1N}} & \frac{1}{c_{2N}} & \ldots & 1 \end{bmatrix} \tag{A.4}$$

where $c_{ij} \in \{1, \ldots, 10\}$ is the result of the comparison question.

It can be shown [142] that the eigen vector of the largest eigen value (which equals N) of W forms a consistent vector of weights to use as weighting functions on the goals.

A.2.2 Marginal Rate of Substitution

The Analytic Hierarchy Process just discussed involved $\frac{N(N-1)}{2}$ questions at each point $d \in$ DPS. This can be simplified to the *marginal rate of substitution* method [160].

In this method, one goal is selected to be compared against, say x_1. Then, $\forall\ j \in \{1, \ldots, N\}$, one asks the designer:

> *How much must the j^{th} goal be increased in preference by amount $\Delta\mu_j$ to compensate for a loss in goal x_1 of an amount $\Delta\mu_1$ in preference?*

Then define

$$\lambda_j = \lim_{\Delta\mu_1 \to 0} \frac{\Delta\mu_j}{\Delta\mu_1} \tag{A.5}$$

and normalizing λ_j by $\sum_{i=1}^{N} \lambda_i$ defines an importance ranking.

A.3 Direct Methods

Rather than invoking such methods for constructing preferences and weights, a design problem may be pursued by simply asking the designer to specify preferences and weights. Such a *direct method* of measurement has historically been shown to suffer from inconsistencies in practice [29, 30]. The results determined may not measure what the modeler thinks is being measured. Other sources may affect the preference rank.

Nonetheless, direct methods appear throughout the literature. Dhingra and Rao freely use such techniques in the construction of preference functions over variables in optimization problems [131, 133, 134, 135], as does Diaz [33, 34, 35]. Harrington also developed his work based on directly assessing preferences [67]. QFD [4, 69] methods directly asks the designer for preference and importance ranks, on a three symbol scale.

Fuzzy set methods are constructed from direct questions on a linguistic scale [42, 204], and indeed Zimmerman shows results can be consistent with experimental verifications [171, 204]. A variety of fuzzy set methods are presented by Dubois and Prade [42] for when X has additional structure beyond a set. The validity of such methods, however, remains theoretically unproven, though the informal arguments for their use presented by Dubois and Prade are persuasive. Dubois and Prade review methods for eliciting "fuzzy membership functions," which can be elicited when X has a known informal structure (such as, for example, "age"), and a known informal context (such as "men"). Then linguistic specifications (such as "very old") can be associated directly with the units of X (such as "60 to 70 years old") via a fuzzy membership function.

The additional elicitation procedure structure needed to form necessary and sufficient conditions for the use of such direct linguistic methods are not known. Clearly, when the additional structure is removed, an intuitive relationship should hold with a lottery method (a lottery method only requires set structure). If the informal arguments for such elicitation procedures are believed, however, they could be used to construct preference functions as used in this thesis, from linguistic terms.

A.4 Difficulties in Determining Preference

This appendix has very briefly discussed methods for eliciting preferences and weights. Even the most theoretically accepted methods, however, do not always assess consistent results [13, 70, 158, 178]. An item initially believed correct may be shown to be inconsistent in subsequent analysis. Thus, as argued in Chapter 6, methods must be developed for *a priori* partial specification.

Some of the techniques presented are only applicable to some problems. For example, both the Analytic Hierarchy Process and the marginal rate of substitution assume an Archimedian property. Thus, non-compensating design strategies will be incompatible with these methods. Further, both techniques ask for amount of increase in some goal relative to amount of increase of others. For theoretical accuracy, the amounts of increase used must limit to zero (Equation A.5). As pointed out by Steuer [160], however, the amounts of increase (the Δ's in Equation A.5) must be large enough for the designer to respond. Thus such techniques will always be, at best, approximations. Again, this reinforces the argument for iteration.

Appendix B

Induced Preference Invariance Proofs

In this appendix, proofs of the claims in Chapter 6 are given. First generalizations of Theorems 1 and 2 in Buckley and Qu [25] are presented. The case of DPS $\simeq \mathbb{R}^n$ and the PPS $\simeq \mathbb{R}$ is considered. Of course, the PPS considered here could be a projection in $\mathbb{R}^m$. Also, $\mu_i : \mathbb{R} \rightarrow [0,1]$ is such that

1. $\text{supp}(\mu) = [d_{min}, d_{max}]$, and

2. there exists $d_* \in (d_{min}, d_{max})$ such that $\mu(d_*) = 1$, and μ is continuous and monotonically increasing on $[d_{min}, d_*]$ and is continuous and monotonically decreasing on $[d_*, d_{max}]$.

Such a μ_i will be termed a *fuzzy number*, in accordance with the fuzzy sets terminology. Also, for the moment, only the case where μ is a preference function on $\mathbb{R}^n$ determined by a non-compensating design strategy, reflected by $\mathcal{P} = min$, will be considered. Also $f : \mathbb{R}^n \rightarrow \mathbb{R}$ will be considered to be an *arbitrary* function. As notation let

$$D^i_\alpha = \{d^i \in \mathbb{R} \mid \mu_i(d^i) \geq \alpha\} \tag{B.1}$$

and let $\text{supp}(\mu)$ denote the support of μ.

Since $\{\mu_i\}$ are fuzzy numbers, D^i_α is a closed interval, say $D^i_\alpha = [d^i_{min}(\alpha), d^i_{max}(\alpha)]$. If $\alpha = 0$, the convention that $D^i_0 = \text{supp}(\mu_i)$ is taken.

Buckley and Qu [25] give what are two equivalent ways to compute the induced preference. One is as given by the extension principle, Definition 6.4, with a non-compensating strategy:

$$\nu(p) = \sup\left\{\min\{\mu_1,\ldots,\mu_n\} \mid d^1,\ldots,d^n \in \mathbb{R}^n : p = f(d^1,\ldots,d^n)\right\} \tag{B.2}$$

and the other is defined as follows

$$\eta(p) = \sup\{\alpha \mid p \in f(D^1_\alpha \times \cdots D^n_\alpha)\} \tag{B.3}$$

One thus wishes to show that

$$\nu(p) = \eta(p) \quad \text{for all} \quad p \in \mathbb{R}. \tag{B.4}$$

On close examination of this statement, it becomes clear that the key ingredient in its proof is the fact that $\nu^{-1}(\alpha) \subsetneq \nu^{-1}(\beta)$ if $\alpha > \beta$. In this case, the key elements of what is needed for the proof of B.4 are contained in the following proposition.

Proposition B.1 *Let* D *be a set, and let* $\{\mathrm{D}_\alpha \subset \mathrm{D} \mid \alpha \in [0,1]\}$ *be a family of subsets which satisfy the inclusion relation* $\mathrm{D}_\alpha \subsetneq \mathrm{D}_\beta$ *for* $\alpha > \beta$. *Let* $\mathrm{f} : \mathrm{D} \to \mathrm{P}$ *be a map of* D *with another set* P. *For* $\mathrm{p} \in \mathrm{P}$ *define*

$$\Theta(\mathrm{p}) = \sup\{\alpha \mid \mathrm{d} \in \mathrm{D}_\alpha, \mathrm{f}(\mathrm{d}) = \mathrm{p}\}$$

and

$$\Phi(\mathrm{p}) = \sup\{\alpha \mid \mathrm{p} \in \mathrm{f}(\mathrm{D}_\alpha)\}$$

Then $\Theta(\mathrm{p}) = \Phi(\mathrm{p})$ *for all* $\mathrm{p} \in \mathrm{P}$.

Proof. The proof follows that in [25] closely. Let $\Phi(p) = \alpha_* > 0$ and let $\epsilon > 0$. Then there exists $\beta \in (\alpha_* - \epsilon, \alpha_*)$. Since

$$\Phi(p) = \sup\{\alpha \mid p \in f(D_\alpha)\}$$

$p \in f(D_\beta)$. Therefore, there exists $d \in D_\beta$ such that $f(d) = p$. Since $D_\beta \subsetneq D_\alpha$ when $\beta < \alpha$ it may be concluded that

$$\Theta(p) = \sup\{\alpha \mid d \in D_\alpha, f(d) = p\} \geq \beta$$

Thus $\Theta(p) \geq \beta \geq \alpha_* - \epsilon$. But, since $\epsilon > 0$ is arbitrary, $\Theta(p) \geq \alpha_* = \Phi(p)$.

Now let $\Theta(p) = \alpha_* > 0$, and let $\epsilon > 0$. As before, there exists $\beta \in (\alpha_* - \epsilon, \alpha_*)$. Since

$$\Theta(p) = \sup\{\alpha \mid d \in D_\alpha, f(d) = p\}$$

and since the inclusion relation holds, one may conclude that $p \in f(D_\beta)$. Thus

$$\Phi(p) = \sup\{\alpha \mid p \in f(D_\alpha)\} \geq \beta$$

so one has $\Phi(p) \geq \beta \geq \alpha_* - \epsilon$. Again, since $\epsilon > 0$ is arbitrary, one has $\Phi(p) \geq \alpha_* = \Theta(p)$.

The last two paragraphs prove that $\Theta(p) = \Phi(p)$ if both quantities are positive. Now if one has $\Theta(p) = 0$ or $\Phi(p) = 0$ then the other is clearly zero since the inclusion relation holds. ■

As a corollary to Proposition B.1 one has

Corollary B.2 *Let* D *and* P *be sets and let* $\mathrm{f} : \mathrm{D} \to \mathrm{P}$ *be a map. Let* $\mu : \mathrm{D} \to [0, 1]$ *be an preference function on* D, *and let* D_α *denote the* α*-cut of* μ. *Suppose that the inclusion relation* $\mathrm{D}_\alpha \subsetneq \mathrm{D}_\beta$ *for* $\alpha > \beta$ *holds. For* $\mathrm{p} \in \mathrm{P}$ *define*

$$\nu(\mathrm{p}) = \sup\{\mu(\mathrm{d}) \mid \mathrm{d} \in \mathrm{D}, \mathrm{f}(\mathrm{d}) = \mathrm{p}\}$$

and

$$\eta(\mathrm{p}) = \sup\{\alpha \mid \mathrm{p} \in \mathrm{f}(\mathrm{D}_\alpha)\}$$

Then $\nu(\mathrm{p}) = \eta(\mathrm{p})$ *for all* $\mathrm{p} \in \mathrm{P}$.

Proof. First note the following

$$\begin{aligned} \nu(p) &= \sup\{\mu(d) \mid d \in D, f(d) = p\} \\ &= \sup\{\alpha \mid d \in D_\alpha, f(d) = p\} \end{aligned}$$

Now, since the inclusion relation holds, the hypotheses of Proposition B.1 are satisfied. ∎

This provides a generalization of Theorem 1 in Buckley and Qu [25] where the special cases $D = \mathbb{R}^n$ and $P = \mathbb{R}$ are considered. It is also possible to generalize Theorem 2 in [25] as follows.

Proposition B.3 *In Proposition 1.1, suppose* D *and* P *are topological spaces, and* f *is a continuous map. Suppose that* $\{\mathrm{D}_\alpha\}$ *are as in Proposition B.1 and that* D_0 *is compact. Let* W_α *and* P_α *be the α-cuts of* Θ *and* Φ*, respectively. Then* $\mathrm{W}_\alpha = \mathrm{P}_\alpha = \mathrm{f}(\mathrm{D}_\alpha)$.

Proof. From Proposition B.1, $W_\alpha = Y_\alpha$. It will be shown that $P_\alpha = f(D_\alpha)$.

Let $p \in f(D_{\alpha_*})$ so that $\Phi(p) = \sup\{\alpha \mid p \in f(D_\alpha)\} \geq \alpha_*$. Thus $p \in P_{\alpha_*}$.

Let $p \in P_{\alpha_*}$ so that $\Phi(p) = \beta_* \geq \alpha_*$. Let $\{\beta_n \in [0,1] \mid n \in \mathbb{Z}\}$ be a sequence which converges to β_* and such that $\beta_n < \beta_*$ for all $n \in \mathbb{Z}$. Since the inclusion relation holds, $y \in f(D_{\beta_n})$. Thus a sequence $\{d_n \in D \mid n \in \mathbb{Z}\}$ can be chosen such that $d_n \in D_{\beta_n}$ and $f(d_n) = p$ for all $n \in \mathbb{Z}$. Since the sequence $\{d_n\}$ lies in D_0 which is compact, there exists a convergent subsequence $\{d_{n_k}\}$ converging to a value, say d_*. Since the sequence $\{\beta_n\}$ converges to β_*, it must be that $d_* \in D_{\beta_*}$, and, since f is continuous, $f(d_*) = p$. Therefore $p \in P_{\alpha_*}$, since $\alpha_* \leq \beta_*$. ∎

Observe that the inclusion $f(D_\alpha) \subseteq P_\alpha = W_\alpha$ always holds.

Proposition B.4 *Let* $\mathcal{P}$ *be a combination function with* μ *the corresponding preference function, and* ν *the preference induced on* $\mathbb{R}$ *by a continuous function* $\mathrm{f} : \mathbb{R}^n \to \mathbb{R}$. *Let* D_α *and* W_α *denote the α-cuts for the preferences* μ *and* ν*, respectively. Then* D_α *and* P_α *are compact for each* $\alpha \in [0,1]$.

Proof. Since $\mathcal{P}$ is a combination function, the support of μ is contained in $D_0^1 \times \cdots \times D_0^n$, the support of *min*. Hence the support of μ is a closed subset of a compact set, and so is compact itself. Now fix $\alpha \in [0,1]$. Then $D_\alpha = \mu^{-1}([\alpha,1])$. Observe that $[\alpha,1]$ is a closed subset of $[0,1]$. Also, μ is continuous since $\mathcal{P}$ and each μ_i are continuous. Thus D_α is closed, and hence compact since it is contained in a compact set. Finally, since f is continuous, it follows that $W_\alpha = f(D_\alpha)$ by Proposition B.1 so W_α is the image of a compact set under a continuous map and so is compact itself. ∎

Proposition B.5 (6.1) *Let $(\mathcal{P},\mathrm{f})$ be strongly inclusive, and let μ be the preference corresponding to $\mathcal{P}$. The preference induced on $\mathbb{R}$ by f is denoted by ν. The α-cut for ν is denoted by W_α. Then $W_\alpha \subseteq P_\alpha$.*

Proof. Let $p \in W_\alpha$ so that $\nu(p) = \beta \geq \alpha$ where $\nu(p) = \sup\{\mu(d) \mid d \in \mathbb{R}^n, f(d) = p\}$.

It is claimed that there exists $d^* \in D_\beta$ such that $f(d^*) = p$. Indeed, since f is continuous, $f^{-1}(p)$ is closed, so $f^{-1}(p) \cap \mathrm{supp}(\mu)$ is compact. Thus μ attains its supremum on $f^{-1}(p) \cap \mathrm{supp}(\mu)$ and this supremum must be β. Thus $d^* \in f^{-1}(p)$ can be chosen to be a point where $\mu(d^*) = \beta$.

Now, since $\mu(d^*) = \beta \geq \alpha$, **H1** implies that $d^* \notin f^{-1}(P^\alpha_{min})$. Therefore $f(d^*) \geq p^\alpha_{min}$. Similarly, by **H2**, $d^* \notin f^{-1}(P^\alpha_{max})$ so $f(d^*) \leq p^\alpha_{max}$. So it has been shown that $f(d^*) = p \in [p^\alpha_{min}, p^\alpha_{max}] = P_\alpha$. ∎

Proposition B.6 (6.2) *Let $(\mathcal{P},\mathrm{f})$ be weakly inclusive, and let W_α be the α-cut for the preference induced on $\mathbb{R}$ by f. Then $\mathrm{P}_\alpha \subseteq \mathrm{W}_\alpha$.*

Proof. Let $p \in P_\alpha$.

It is claimed that there exists an $d \in D^1_\alpha \times \cdots \times D^n_\alpha$ such that $f(d) = p$. Indeed, all the ridge points are contained in $D^1_\alpha \times \cdots \times D^n_\alpha$ and therefore, in particular, d^α_{min} and d^α_{max} are so contained. Also note that $f(d^\alpha_{min}), f(d^\alpha_{max}) \in P_\alpha$. Since f is continuous and $D^1_\alpha \times \cdots \times D^n_\alpha$ is connected, $f(D^1_\alpha \times \cdots \times D^n_\alpha)$ is connected. Hence it must be either an open, closed, or half-open interval containing p^α_{min} and p^α_{max}. In any case it must be that $P_\alpha \subseteq f(D^1_\alpha \times \cdots \times D^n_\alpha)$ and so there is indeed an $d \in D^1_\alpha \times \cdots \times D^n_\alpha$ such that $f(d) = p$.

If D_α is the α-cut for the preference μ on $\mathbb{R}^n$, then Proposition 4.1 implies that $D^1_\alpha \times \cdots \times D^n_\alpha \subseteq D_\alpha$. Therefore, the d found above must lie in D_α. Now, since $f(d) = p$ and $\mu(d) \geq \alpha$, $\nu(p) \geq \alpha$, hence $p \in W_\alpha$. ∎

Proposition B.7 (6.3) *Let $\mathcal{P}$ be a combination function and let* $\mathrm{f} : \mathbb{R}^n \rightarrow \mathbb{R}$ *be a function. If* W_α *is the α-cut for the preference induced on* $\mathbb{R}$ *by* f*, and if* $\mathrm{W}_\alpha = \mathrm{P}_\alpha$ *then the pair* $(\mathcal{P}, \mathrm{f})$ *satisfies* **H1** *and* **H2**.

Proof. Suppose that $W_\alpha = P_\alpha$ and that **H1** does not hold. Then there exists $d \in D_\beta \cap f^{-1}(P^\alpha_{min})$ for some $\beta \geq \alpha$. Thus $\nu(f(d)) \geq \beta$ and $f(d) \in W_\beta \subseteq W_\alpha$. But, since $W_\alpha = P_\alpha$, $f(d) \geq p^\alpha_{min}$. This contradicts $d \in f^{-1}(P^\alpha_{min})$, therefore **H1** must hold.

Now suppose that $W_\alpha = P_\alpha$ and that **H2** does not hold. Then there exists $d \in D_\beta \cap f^{-1}(P^\alpha_{max})$ for some $\beta \geq \alpha$. Thus $\mu(d) \geq \beta$ so $\nu(f(d)) \geq \beta$, hence $f(d) \in Y_\alpha$ since $P_\beta = W_\beta \subseteq W_\alpha = P_\alpha$. Therefore $f(d) \leq y^\alpha_{max}$, contradicting $d \in f^{-1}(P^\alpha_{max})$. Thus **H2** must hold. ∎

Appendix C

Nomenclature

A very brief explanation of the symbols and operations used are given. Halmos [65], Royden [139], and Lang [89] provide further reading.

empty set: The empty set $\emptyset$ is a set with no elements. For example, $\{x \in \mathbb{R} \mid x^2 = -1\} = \emptyset$.

composition: The composition $\circ$ is used to mean

$$\mu \circ f(d) = \mu(f(d))$$

strict subset: The strict subset relation $A \subsetneq B$ indicates A is a subset of B, but $A \neq B$. For example, $(a, b) \subsetneq [a, b]$, where $a < b \in \mathbb{R}$.

image: The image of a point $d \in D$ through a map $f : D \to P$ is/are the value(s) $f(d)$. The image of a set $\{d\} \subset D$ through a map $f : D \to P$ is/are the value(s) $\{p \in P \mid p = f(\mathfrak{d}), \mathfrak{d} \in \{d\}\}$. The image might be $\emptyset$.

pre-image: The pre-image of a value $p \in P$ through a map $f : D \to P$ is/are the point(s) $\{d \in D \mid p = f(d)\}$. The pre-image of a set $\{p\} \subset P$ through a map $f : D \to P$ is/are the point(s) $\{d \in D \mid \mathfrak{p} = f(d), \mathfrak{p} \in \{p\}\}$. The pre-image might be $\emptyset$.

sup: The *sup* of a function $f : D \to P \subset \mathbb{R}$ over a set D is the least upper bound of $f(d)$ in $\mathbb{R}$, $d \in D$.

inf: The *inf* of a function $f : D \to P \subset \mathbb{R}$ over a set D is the greatest lower bound of $f(d)$ in $\mathbb{R}$, $d \in D$.

topology: A topology on a set S is a collection $\mathcal{O}$ of subsets, called *open*, restricted to satisfy, for all $A_i \in \mathcal{O}$:

$$\begin{aligned} i) &\quad \emptyset \in \mathcal{O},\ S \in \mathcal{O} \\ ii) &\quad A_1 \cap A_2 \in \mathcal{O} \\ iii) &\quad \cup_{i=1}^{\infty} A_i \in \mathcal{O} \end{aligned}$$

For example, the usual topology on $\mathbb{R}$ is the collection of all subsets which are unions of open intervals. A *topological space* is a pair $(S, \mathcal{O})$.

σ-algebra: A σ-algebra of sets is a collection $\mathcal{B}$ of subsets $\{N_j\}$ of a set S such that

$$\begin{aligned} i) &\quad \emptyset \in \mathcal{B},\ S \in \mathcal{B} \\ ii) &\quad \neg N_j = S \backslash N_j \in \mathcal{B} \\ iii) &\quad \cup_{j=1}^{\infty} N_j \in \mathcal{B} \end{aligned}$$

This, together with DeMorgan's laws, is sufficient to show the collection forms an algebra under union and intersection. A *measurable space* is a pair $(S, \mathcal{B})$. For example, the usual measurable space on $\mathbb{R}$ is the collection of all subsets which are unions of half-open intervals, closed on the left. Given a measurable space, there is a largest topological space contained within. Given a topological space, there is a smallest measurable space containing it.

projection: Consider a vector space V over $\mathbb{R}$ (more generally over a field F), where $v \in V$ has components $(v^1, \ldots, v^n)$ in some basis. The projection on the i$^{\text{th}}$ component is a map

$$\begin{aligned} \pi_i \quad : \quad & V \to \mathbb{R} \\ & v = (v^1, \ldots, v^n) \mapsto v^i \end{aligned}$$

manifold: A smooth manifold is a set S with an equivalence class of atlases $\mathcal{A}$, where an atlas is a family of charts $\phi : U \to \phi(U)$, where U is an open subset of S, and $\phi(U)$ is an open subset of $\mathbb{R}^n$. Manifolds are subsets of $\mathbb{R}^N$ which locally "look like" $\mathbb{R}^n$, $n < N$. For example, the 2-sphere $\mathbb{S}^2 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}$ is a manifold in $\mathbb{R}^3$, which locally "looks like" $\mathbb{R}^2$.

vector bundle: A vector bundle E is a manifold B and associated to each $b \in B$ is a vector space E_b (for example, $E_b \simeq \mathbf{R}^d$), where the set $E = \cup_{b \in B} E_b$ is a manifold. The map $\Pi : E \to B : e \in E_b \mapsto b$ is the vector bundle projection.

section: A section of a vector bundle E is a map $\xi : B \to E$ such that $\Pi \circ \xi = \mathrm{id}_B$.

tangent bundle: A tangent bundle TM is a manifold M and associated to each $m \in M$ is the set of equivalence classes of curves tangent to M passing through m (the tangent space at m). The tangent bundle of a manifold is the collection of points in the manifold along with the best linear approximation to the manifold at each point. The tangent bundle is a vector bundle.

vector field: A vector field X is a section of the tangent bundle TM. Thus

$$X : M \to TM$$

A vector field assigns to each point in M a tangent vector.

flow: A family of diffeomorphisms $\{F_t\}$, $F_t : M \to M$, is a flow on M provided F_0 is the identity, and $F_{t+s} = F_t F_s$. Using a complete vector field X, a flow F_t can be defined as the integral curves of X. That is, the flow and the vector field are related by

$$\frac{dF_t(m)}{dt} = X(F_t(m))$$

References

[1] *Improving Engineering Design: Designing for Competitive Advantage.* National Academy Press, Washington, D.C., 1990. National Research Council: Committee on Engineering Design Theory and Methodology, Manufacturing Studies Board, Commission on Engineering and Technical Systems.

[2] J. L. Adams. *Conceptual Blockbusting.* W. H. Freeman and Company, San Francisco, Ca., 1974.

[3] P. Adby and M. Dempster. *Introduction to Optimization Methods.* Chapman and Hall, London, 1974.

[4] Y. Akao. *Quality Function Deployment: Integrating Customer Requirements into Product Design.* Productivity Press, Cambridge, 1990.

[5] J. Alger and C. Hays. *Creative Synthesis in Design.* Prentice Hall, Englewood Cliffs, 1964.

[6] J. S. Arora. *Introduction to Optimum Design.* McGraw-Hill Book Company, New York, 1989.

[7] K. Arrow. *Social Choice and Individual Values.* J. Wiley, 1st edition, 1951.

[8] S. Ashley. Engineous explores the design space. *Mechanical Engineering,* pages 49–52, February 1992.

[9] M. Asimow. *Introduction to Design.* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1962.

[10] S. Baas and H. Kwakernaak. Rating and ranking of multiple-aspect alternatives using fuzzy sets. *Automatica*, 13:47–48, 1977.

[11] A. Baker. Cars of future court consumer lifestyles. *Design News*, 48(4):23–24, 1992.

[12] T. Barker. *Quality by Experimental Design.* Marcel Dekker, Inc., New York, 1985.

[13] D. Bell, H. Raiffa, and A. Tversky. Descriptive, normative, and prescriptive interactions in decision making. In D. Bell, H. Raiffa, and A. Tversky, editors, *Decision Making*, pages 9–30, Cambridge, 1988. Cambridge University Press.

[14] R. E. Bellman and M. Giertz. On the analytic formalism of the theory of fuzzy sets. *Information Sciences*, 5:149–156, 1973.

[15] R. E. Bellman and L. A. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 27(4), December 1970.

[16] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev. Linear programming with multiple objective functions: Step method (STEM). *Mathematical Programming*, 1(3):366–375, 1971.

[17] P. Bernays. *Axiomatic Set Theory.* North-Holland, New York, 1968.

[18] P. Biegel and M. Pecht. Design trade-offs made easy. *Concurrent Engineering*, 1(3):29–40, May/June 1991.

[19] C. Boender, J. de Graan, and F. Lootsma. Multi-criteria decision analysis with fuzzy pairwise comparisons. *Fuzzy Sets and Systems*, 29:133–143, 1989.

[20] G. E. Box. *Statistics for Experimenters.* J. Wiley and Sons, New York, 1978.

[21] G. E. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis.* Addison-Wesley Publishing Co., Reading, MA, 1973.

[22] D. Brown. Capturing mechanical design knowledge. In *Proceedings of the 1985 Computers in Engineering Conference*, pages 121–129, Boston, 1985. ASME.

[23] J. Buckley. Decision making under risk: A comparison of Bayesian and fuzzy set methods. *Risk Analysis*, 3(3):157–168, 1983.

[24] J. Buckley. Fuzzy heirarchical analysis. *Fuzzy Sets and Systems*, 17:233–247, 1985.

[25] J. Buckley and Y. Qu. On using α-cuts to evaluate fuzzy equations. *Fuzzy Sets and Systems*, 38:309–312, 1990.

[26] D. M. Byrne and S. Taguchi. The Taguchi approach to parameter design. In *Quality Congress Transaction - Anaheim*, pages 168–177. ASQC, May 1986.

[27] G. Chamberlain. Good cancels bad in auto noise. *Design News*, pages 128–132, October 1991.

[28] R. T. Cox. *The Algebra of Probable Inference.* Johns-Hopkins University Press, Baltimore, MD, 1961.

[29] L. Cronbath. Response sets and test validity. *Educ. Psychol. Measmt.*, 6:475–495, 1946.

[30] L. Cronbath. Further evidence on response sets and test designs. *Educ. Psychol. Measmt.*, 10:3–31, 1950.

[31] A. Dempster. Upper and lower probabilities induced by a multivalued map. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[32] V. Dem'yanov and V. Malozemov. *Introduction to Minimax.* Dover Publications, New York, 1974.

[33] A. R. Diaz. Fuzzy set based models in design optimization. In S. S. Rao, editor, *Advances in Design Automation - 1988*, volume DE-14, pages 477–485, New York, September 1988. ASME.

[34] A. R. Diaz. Modeling of computer-based decisions and optimization in the design process. In S. L. Newsome and W. R. Spillers, editors, *Design Theory '88*, pages 286–295, RPI, Troy, NY, June 1988. NSF. 1988 NSF Grantee Workshop on Design Theory and Methodology.

[35] A. R. Diaz. A strategy for optimal design of hierarchial systems using fuzzy sets. In J. R. Dixon, editor, *The 1989 NSF Engineering Design Research Conference*, pages 537–547, College of Engineering, University of Massachusetts, Amherst, June 1989. NSF.

[36] J. R. Dixon. Iterative redesign and respecification: Research on computational models of design processes. In S. L. Newsome and W. R. Spillers, editors, *Design Theory '88: Proceedings of the 1988 Grantee Workshop on Design Theory and Methodology*, pages 3.7.1–3.7.13, RPI, Troy, NY, June 1988. NSF. In the workshop proceedings (preprints) only.

[37] D. C. Dlesk and J. S. Liebman. Multiple objective engineering optimization. *Engineering Optimization*, 6:161–175, 1983.

[38] W. Dong and H. Shaw. Vertex method for computing functions on fuzzy variables. *Fuzzy Sets and Systems*, 24(1):65–78, 1987.

[39] 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.

[40] J. Doyle, B. Francis, and A. Tannenbaum. *Feedback Control Theory.* MacMillan Publishing Co., 1990.

[41] D. Dubois and H. Prade. Fuzzy real algebra: Some results. *Fuzzy Sets and Systems*, 2:327–348, 1979.

[42] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications.* Academic Press, New York, 1980.

[43] D. Dubois and H. Prade. A class of fuzzy measures based on triangular norms. *International Journal of General Systems*, 8:43–61, 1982.

[44] D. Dubois and H. Prade. Towards fuzzy differential calculus, part 1: Integration of fuzzy mappings. *Fuzzy Sets and Systems*, 8:1–17, 1982.

[45] D. Dubois and H. Prade. Towards fuzzy differential calculus, part 2: Integration of fuzzy intervals. *Fuzzy Sets and Systems*, 8:105–116, 1982.

[46] D. Dubois and H. Prade. Towards fuzzy differential calculus, part 3: Differentiation. *Fuzzy Sets and Systems*, 8:225–253, 1982.

[47] D. Dubois and H. Prade. Criteria aggregation and ranking of alternatives in the framework of fuzzy sets. In *TIMS/Studies in the Management Sciences*, pages 209–240. Elsevier Science Publishers B.V., North-Holland, 1984.

[48] D. Dubois and H. Prade. Fuzzy numbers: An overview. In *Analysis of Fuzzy Information*, volume 1, pages 3–39, Boca Raton, FL, 1987. CRC Press.

[49] D. Dubois and H. Prade. Decision evaluation methods under uncertainty and imprecision. In J. Kacprzyk and M. Fedrizzi, editors, *Combining Fuzzy Imprecision with Probabilistic Uncertainty in Decision Making*, Heidelberg, 1988. Springer Verlag. Springer Lecture Notes in Economics and Mathematical Systems, Vol. 310.

[50] D. Dubois and H. Prade. *Possibility Theory: An Approach to the Computerized Processing of Information.* Plenum Press, New York, 1988.

[51] J. Efstathiou and Rajkovič. Multiattribute decisionmaking using a fuzzy heuristic approach. *IEEE Transactions on Systems, Man and Cybernetics*, 9(6):326–333, 1979.

[52] H. Eschenauer, J. Koski, and A. Osyczka, editors. *Multicriteria Design Optimization.* Springer-Verlag, Berlin, 1990.

[53] P. Fishburn. *Foundations of Expected Utility.* D. Reidel, 1982.

[54] T. Fowler. *Value Analysis in Design.* Competitive Manufacturing Series. Van Nostrand Reinhold, New York, 1990.

[55] T. Freiheit and S. S. Rao. A modified game theory approach to multi-objective optimization. In S. S. Rao, editor, *Advances in Design Automation - 1988*, volume DE-14, pages 107–114, New York, September 1988. ASME.

[56] M. J. French. *Conceptual Design for Engineers.* Springer-Verlag, London, 1985.

[57] S. French. Fuzzy decision analysis: Some criticisms. In *TIMS/Studies in the Management Sciences*, pages 29–44. Elsevier Science Publishers B.V., North-Holland, 1984.

[58] S. French. *Decision Theory: An Introduction to the Mathematics of Rationality.* Ellis Horwood, Chichester, 1988.

[59] L. Fung and K. Fu. An axiomatic approach to rational decision making in a fuzzy environment. In L. Zadeh et al., editors, *Fuzzy Sets and Their Applications to Cognitive and Design Processes*, New York, 1974. Acedemic Press.

[60] A.M. Geoffrion, J.S. Dyer, and A. Feinberg. An interactive approach for multicriterion optimization, with an application to the operation of an academic department. *Management Science*, 19(4):357–368, 1972.

[61] R. Goetschel and W. Voxman. Elementary fuzzy calculus. *Fuzzy Sets and Systems*, 18:31–43, 1986.

[62] B. Golden, E. Wasil, and P. Harker, editors. *The Analytic Heirarchy Process.* Springer-Verlag, New York, 1989.

[63] M. Grabish, M. Yoneda, and S. Fukami. Subjective evaluation by fuzzy integral: the crisp and possibilistic case. In T. Terano et al., editors, *Fuzzy Engineering toward Human Friendly Systems*, pages 33–41, Yokohama JP, November 1991. LIFE, IFES.

[64] P. Halmos. *Measure Theory.* Springer-Verlag, New York, 1974.

[65] P. Halmos. *Naive Set Theory.* Springer-Verlag, New York, 1974.

[66] J. Hammersley and D. Handscomb. *Monte Carlo Methods.* Methuen, London, 1964.

[67] E. Harrington. The desirability function. *Industrial Quality Control*, pages 494–498, 1965.

[68] M. J. Harry. The nature of six sigma quality. Technical report, Government Electronics Group, Motorola, Inc., 1991.

[69] J. Hauser and D. Clausing. The house of quality. *Harvard Business Review*, pages 63–73, May/June 1988.

[70] J. Hershey, H. Kunreuther, and P. Schoemaker. Sources of bias in assessment procedures for utility functions. In D. Bell, H. Raiffa, and A. Tversky, editors, *Decision Making*, pages 422–442, Cambridge, 1988. Cambridge University Press.

[71] B. Hollins and S. Pugh. *Sucessful Product Design.* Butterworths, London, 1990.

[72] A. Howe, J. Dixon, et al. Dominic: A domain independent program for mechanical engineering design. In *Applications of Artificial Intelligence in Engineering Problems*, pages 290–300, U.K., April 1986. Southampton University.

[73] J.S. Hunter. Statistical design applied to product design. *Journal of Quality Technology*, 17(4):210–221, October 1985.

[74] K. Ishii and M. Sugeno. A model of human evaluation process using fuzzy measure. *International Journal of Man-Machine Studies*, 22:19–38, 1985.

[75] G. R. Iversen. *Bayesian Statistical Inference.* Series no. 07-043. Sage Publications, Beverly Hills and London, 1984.

[76] P. Jain and A. Agogino. Calibration of fuzzy linguistic variables for expert systems. In *Proc. of the ASME Computers in Engineering Conference, Volume I,* pages 313–318, San Francisco, CA, July 1988. ASME.

[77] R. Jain. Decisionmaking in the presence of fuzzy variables. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 698–703, 1976.

[78] R. Jain. A procedure for multiple-aspect decision making using fuzzy sets. *International Journal of System Science*, 8(1):1–7, 1977.

[79] H. Jeffreys. *Theory of Probability.* Clarendon Press, third edition, 1961.

[80] P. W. John. *Statistical Design and Analysis of Experiments.* Macmillan Co., 1971.

[81] R. N. Kackar. Off-line quality control, parameter design, and the Taguchi approach. *Journal of Quality Technology*, 17(4), October 1985.

[82] O. Kaleva. Fuzzy differential equations. *Fuzzy Sets and Systems*, 24:301–317, 1987.

[83] A. Kaufmann and M. M. Gupta. *Introduction to Fuzzy Arithmetic: Theory and Applications*. Electrical/Computer Science and Engineering Series. Van Nostrand Reinhold Company, New York, 1985.

[84] A. Kaufmann and M. M. Gupta. *Fuzzy Mathematical Models in Engineering and Management Science*. North Holland, Amsterdam, 1988.

[85] G. J. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[86] P. Korhonen and J. Laasko. A visual interactive method for solving the multiple criteria problem. *European Journal of Operations Research*, 1985.

[87] D. Krantz, R. Luce, P. Suppes, and A. Tversky. *Foundations of Measurement*, volume I. Academic Press, New York, 1971.

[88] W. L. Kubic and F. P. Stein. A theory of design reliability using probability and fuzzy sets. *AIChE Journal*, 34(4):583–601, April 1988.

[89] S. Lang. *Differential Manifolds*. Springer-Verlag, New York, 1985.

[90] W. Lewis and A. Samuel. *Fundamentals of Engineering Design: Ideads, Methods, and Applications*. Prentice-Hall, 1989.

[91] A. Lucas. Motorola's kaizen brigrade. *Design News*, 48(3):98–101, February 1992.

[92] M. Luhandjula. Fuzzy optimization: An appraisal. *Fuzzy Sets and Systems*, 30:257–282, 1989.

[93] T. Lynch. Highlights from the 1992 SAE international congress and exhibition. *Design News*, 48(6):22, 1992.

[94] K. Menger. Statistical metrics. *Proceedings of the National Academy of Sciences*, 28:535–537, 1942.

[95] R. Meyer and J. Pratt. The consistent assessment and fairing of preference functions. *IEEE Transactions on System Science and Cybernetics*, 4(3):270–278, September 1968.

[96] J. Milnor. Games against nature. In R. Thrall et al., editors, *Decision Processes*, pages 49–59, New York, 1957. J. Wiley and Sons.

[97] M. Mizumoto and K. Tanaka. Some properties of fuzzy numbers. In M.M. Gupta et. al., editor, *Adavances in Fuzzy Sets: Theory and Applications*, Amsterdam, 1984. North Holland.

[98] D. C. Montgomery. *Design and Analysis of Experiments.* Wiley, New York, 1991.

[99] R. E. Moore. *Methods and Applications of Interval Analysis.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[100] J. Mostow. Towards better models of the design process. *The AI Magazine*, pages 44–57, Spring 1985.

[101] Y. Nakamura. Extension of algebraic calculus on fuzzy numbers using alpha-level sets. In *Fuzzy Information Processing Symposium (FIP-84)*, Kauai, Hawaii, 1984.

[102] C. V. Negoiţă and D. A. Ralescu. *Applications of Fuzzy Sets to Systems Analysis.* Halsted Press, New York, 1975.

[103] H. T. Nguyen. A note on the extension principle for fuzzy sets. *Journal of Math. Anal. Appl.*, 64:369–380, 1978.

[104] Y. Nishiwaki. Fuzzy optimization of radiation protection and nuclear safety. In J. Kacprzyk and M. Fedrizzi, editors, *Combining Fuzzy Imprecision with Probabilistic Uncertainty in Decision Making*, pages 353–273, New York, 1988. Spring-Verlag.

[105] Vilém Novák. *Fuzzy Sets and Their Applications.* Adam Hilger, Philadelphia, 1989. Published in English by IOP Publishing Ltd.

[106] A. Osycska. *Multi-Criterion Optimization in Engineering with FORTRAN Examples.* Halstad Press, New York, 1984.

[107] K. Otto, A. Lewis, and E. Antonsson. Approximating α-cuts with a ridge point algorithm. *Fuzzy Sets and Systems*, 1992. Submitted for review. EDRL Technical Report 91j.

[108] K. Otto, A. Lewis, and E. Antonsson. Determining optimal preference points with dependent variables. *Fuzzy Sets and Systems*, 1992. Submitted for review. EDRL Technical Report 91k.

[109] K. N. Otto and E. K. Antonsson. Combing the Functional Requirement and the Design Parameter's Specification to Resolve a Performance Parameter Value. Engineering Design Research Laboratory Report EDRL-TR 89d, California Institute of Technology, Pasadena, CA 91125, September 1989.

[110] K. N. Otto and E. K. Antonsson. The Behaviour of Fuzzy and Probabilistic Calculations. Engineering Design Research Laboratory Report EDRL-TR 90a, California Institute of Technology, Pasadena, CA 91125, 1990.

[111] K. N. Otto and E. K. Antonsson. The Taguchi Method of Product Design. Engineering Design Research Laboratory Report EDRL-TR 90c, California Institute of Technology, Pasadena, CA 91125, September 1990.

[112] K. N. Otto and E. K. Antonsson. Design Parameter Selection in the Presence of Noise. In *Proceedings of the 1992 ASME DTM Conference*, Scottsdale, AZ, 1991. ASME. Accepted for publication. EDRL Technical Report 90g.

[113] K. N. Otto and E. K. Antonsson. Extensions to the Taguchi Method of Product Design. *ASME Journal of Mechanical Design*, 1991. Accepted for publication. EDRL-TR 90f. Also appears in the *Proceedings of the 1991 ASME DTM Conference, DE-Vol. 31*, pages 21-30, Miami FL.

[114] K. N. Otto and E. K. Antonsson. Operator Equations under Imprecision. Engineering Design Research Laboratory Report EDRL-TR 91d, California Institute of Technology, 1991.

[115] K. N. Otto and E. K. Antonsson. Representing and Manipulating Uncertainties in Preliminary Engineering Design. In *1991 NSF Design and Manufacturing Systems Grantees Conference*. The University of Texas at Austin, January 1991.

[116] K. N. Otto and E. K. Antonsson. Some α-cut Concepts. Engineering Design Research Laboratory Report EDRL-TR 91c, California Institute of Technology, 1991.

[117] K. N. Otto and E. K. Antonsson. Trade-Off Strategies in Engineering Design. *Research in Engineering Design*, 3(2):87–104, 1991.

[118] K. N. Otto and E. 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.

[119] K. N. Otto and E. K. Antonsson. Tuning Parameters in Engineering Design. *ASME Journal of Mechanical Design*, 1991. Accepted for publication. EDRL-TR 91a. Also appears in the *Proceedings of the 1991 ASME DTM Conference, DE-Vol. 31*, pages 37-44, Miami FL.

[120] K. N. Otto, A. D. Lewis, and E. K. Antonsson. Fuzzy Preference Induced on Manifolds by Vector Fields and Flows. Engineering Design Research Laboratory Report EDRL-TR 91f, California Institute of Technology, 1991.

[121] G. Pahl and W. Beitz. *Engineering Design.* The Design Council, Springer-Verlag, New York, 1984.

[122] M. Palmer. *The Architect's Guide to Facility Programming.* American Institute of Architects, Washington, D.C., 1981.

[123] P. Papalambros and D. Wilde. *Principles of Optimal Design.* Cambridge University Press, New York, 1988.

[124] M. Phadke. *Quality Engineering Using Robust Design.* Prentice Hall, Englewood Cliffs, NJ, 1989.

[125] D. Pierre. *Optimization Theory with Applications.* Dover Press, New York, 1986.

[126] S. Pugh. *Total Design.* Addison-Wesley, New York, 1990.

[127] M. Puri and D. A. Ralescu. Differentials of fuzzy functions. *Journal of Mathematical Analysis and Applications*, 91:552–558, 1983.

[128] R. Putrus. Non-traditional approach in justifying computer integrated manufacturing systems. In *Autofact '89 Conference Proceedings*, pages 8.1–8.26, Detroit, MI, October 1989. SME.

[129] H. Raiffa and R. Keeney. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York, 1976.

[130] R. Ramaswamy, K. Ulrich, N. Kishi, and M. Tomikashi. Solving parametric design problems requiring configuration choices. In L. Stauffer, editor, *Advances in Design Automation*, pages 103–110, New York, September 1991. ASME. Volume DE-31.

[131] S. S. Rao. Description and optimum design of fuzzy mechanical systems. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, pages 1–7, 1986.

[132] S. S. Rao. Description and optimum design of fuzzy mechanical systems. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 109:126–132, March 1987.

[133] S. S. Rao and A. K. Dhingra. Integrated optimal design of planar mechanisms using fuzzy theories. In *Advances in Design Automation - 1989*, volume DE-15-2, pages 161–168, New York, September 1989. ASME.

[134] S. S. Rao and A. K. Dhingra. Applications of fuzzy theories to multi-objective system optimization. NASA Contractor Report 177573, NASA Ames Research Center, Moffett Field, CA 94035-1000, 1991. University Consortium Interchange Number NCA2-223.

[135] S. S. Rao, A. K. Dhingra, and V. Kumar. Nonlinear membership functions in the fuzzy optimization of mechanical and structural systems. Technical Report 90-1175-CP, AIAA, New York, 1990.

[136] S. S. Rao, K. Sundararaju, B. Prakash, and C. Balakrishna. Multiobjective fuzzy optimization techniques for engineering design. *Computers and Structures*, 42(1):37–44, 1992.

[137] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell. *Engineering Optimization, Methods and Applications*. Wiley, New York, 1983.

[138] P. Ross. *Taguchi Techniques for Quality Engineering*. McGraw Hill, New York, 1988.

[139] H. L. Royden. *Real Analysis*. Macmillan, New York, 1988.

[140] T. Saaty. Exploring the interface between hierarchies, multiple objectives, and fuzzy sets. *Fuzzy Sets and Systems*, 1:57–68, 1978.

[141] T. Saaty. *The Analytic Heirarchy Process*. McGraw Hill, New York, 1980.

[142] T. Saaty. Axiomatic foundation of of the analytic heirarchy process. *Management Science*, 32:841–855, 1986.

[143] M. Sakawa. Interactive multiobjective decision-making by the fuzzy sequential proxy optimization technique – FSPOT. In *TIMS/Studies in the Management Sciences*. Elsevier Science Publishers B.V., North-Holland, 1984.

[144] M. Sakawa and H. Yano. Interactive decision making for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 29:315–326, 1989.

[145] M. Sakawa and H. Yano. Interactive fuzzy decision making for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 32:245–261, 1989.

[146] M. Sakawa and H. Yano. An interactive fuzzy satisficing method for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 30:221–238, 1989.

[147] M. Sakawa and H. Yano. Feasibility and Pareto optimality for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems*, 43:1–15, 1991.

[148] M. Salukvadze. *Vector-Valued Optimization Problems in Control Theory*. Academic Press, New York, 1979.

[149] E. Sandgren. A multi-objective design tree approach for the optimization of mechanisms. *Mechanisms and Machine Theory*, 25(3):257–272, 1990.

[150] J. Savage. *The Foundations of Statistics*. (Revised Edition, Dover Publication, 1972). Wiley, New York, 1954.

[151] R. Schank. Where's the AI? *The AI Maganzine*, 12(4):38–49, 1991.

[152] S. Seikkala. On the fuzzy initial value problem. *Fuzzy Sets and Systems*, 24:319–330, 1987.

[153] J. Seinfeld and W. McBride. Optimization with multiple performance criteria. *Ind. Eng. Chem. Process Des. Develop.*, 9(1):53–57, 1970.

[154] S. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[155] J. N. Siddall. *Analytical Decision Making in Engineering Design*. Prentice-Hall, 1972.

[156] J. N. Siddall. *Probabilistic Engineering Design; Principles and Applications*. Marcel Dekker, New York, 1983.

[157] N. D. Singpurwalla. Design by decision theory: A unifying perspective on Taguchi's approach to quality engineering. In *NSF Design and Manufacturing Systems Grantees Conference*, Tempe Arizona, 1990. NSF. In the supplement to the Proceedings.

[158] P. Slovic, B. Fischhoff, and S. Lichtenstein. Response model, framing, and information processing effects in risk assessment. In D. Bell, H. Raiffa, and A. Tversky, editors, *Decision Making*, pages 152–166, Cambridge, 1988. Cambridge University Press.

[159] P. Smets. Belief functions. In P. Smets, E. Mamdani, D. Dubois, and H. Prade, editors, *Non-Standard Logics for Automated Reasoning*, pages 254–286. Academic Press, 1988.

[160] R. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application.* J. Wiley, New York, 1986.

[161] G. Stiny. Computing with form and meaning in architecture. UCLA, Grad. School of Arch. and Urban Planning, 1988.

[162] G. Stiny. Formal devices for design. UCLA, Grad. School of Arch. and Urban Planning, 1988.

[163] G. Stiny. Weights. UCLA, Grad. School of Arch. and Urban Planning, 1991.

[164] M. Sugeno. *Theory of fuzzy integrals and its applications.* Ph.D. thesis, Tokyo Institute of Technology, 1974.

[165] M. Sugeno. Fuzzy measures and fuzzy integrals – a survey. In M. Gupta et al., editors, *Fuzzy Automata and Decision Processes*, pages 89–102, New York, 1977. North-Holland.

[166] N. P. Suh. *The Principles of Design.* Oxford University Press, New York, 1990.

[167] P. Suppes. *Axiomatic Set Theory.* Dover Publications, New York, 1972.

[168] A. Sveshnikov. *Problems in Probability Theory, Mathematical Statistics, and the Theory of Random Functions.* Dover Publications, New York, 1968.

[169] G. Taguchi. *Introduction to Quality Engineering.* Asian Productivity Organization, Unipub, White Plains, NY, 1986.

[170] G. Taguchi and M. Phadke. Quality engineering through design optimization. *IEEE*, pages 1106–1113, 1984.

[171] U. Thole, H. Zimmerman, and P. Zysno. On the suitability of minimum and product operators for the intersection of fuzzy sets. *Fuzzy Sets and Systems*, 2(2):167–180, 1979.

[172] D. Thurston and J. Carnahan. Fuzzy ratings and utility analysis in preliminary design: evaluation of multiple attributes. *ASME Journal of Mechanical Design*, September 1990. Submitted for review.

[173] D. Thurston and Y. Tian. A method for integrating utility analysis into an expert system for design evaluation under uncertainty. In B. D'Ambrosio, editor, *Proceedings of the Seventh Uncertainty in Artificial Intelligence Conference*, pages 398–405, San Mateo, CA, 1991. Morgan Kaufmann Publishers.

[174] T. Tomiyama and H. Yoshikawa. Extended general design theory. In H. Yoshikawa and E. Warman, editors, *Design Theory for CAD*, pages 95–130. IFIP, Elsevier Science Publishers, 1987.

[175] E. Triantaphyllou and S. Mann. An evaluation of the eigenvalue approach for determining the membership values in fuzzy sets. *Fuzzy Sets and Systems*, 35:295–301, 1990.

[176] M. Tribus. *Rational Descriptions, Decisions, and Designs.* Pergamon Press, New York, 1969.

[177] S. Tsai and K. Ragsdell. Orthogonal arrays and conjugate directions for Taguchi class optimization. In S. S. Rao, editor, *Proccedings of the 1988 Design Automation Conference*, 1988.

[178] A. Tversky and D. Kahneman. Rational choice and the framing of decisions. In D. Bell, H. Raiffa, and A. Tversky, editors, *Decision Making*, pages 167–192, Cambridge, 1988. Cambridge University Press.

[179] D. G. Ullman. *The Mechanical Design Process.* McGraw Hill, New York, 1992.

[180] K. Ulrich, D. Sartorius, S. Pearson, and M. Jakelia. Including the value of time in design-for-manufacturing decision making. Technical report, MIT, 1990.

[181] T. Vincent. Game theory as a design tool. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 105:165–170, June 1983.

[182] A. Ward. *A Theory of Quantitative Inference for Artifact Sets, Applied to a Mechanical Design Compiler.* Ph.D. thesis, MIT, 1989.

[183] A. Ward. A recursive model for managing the design process. In *Proccedings of the 1990 Design Theory and Methodology Conference*, pages 47–52, New York, 1990. ASME.

[184] A. C. Ward and W. P. Seering. Extending the constraint propogation of intervals. *Artificial Intelligence in Engineering Design and Manufacturing*, 4(1):47–51, 1990.

[185] I. Wilson and M. Wilson. *From Idea to Working Model.* J. Wiley and Sons, New York, 1970.

[186] R. L. Winkler. *An Introduction to Bayesian Inference and Decision.* Holt, Rinehart and Winston Inc., 1972.

[187] K. L. Wood. *A Method for Representing and Manipulating Uncertainties in Preliminary Engineering Design.* Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1989.

[188] K. L. Wood and E. 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.

[189] K. L. Wood and E. K. Antonsson. Modeling Imprecision and Uncertainty in Preliminary Engineering Design. *Mechanism and Machine Theory*, 25(3):305–324, February 1990.

[190] K. L. Wood, E. K. Antonsson, and J. L. Beck. Representing Imprecision in Engineering Design – Comparing Fuzzy and Probability Calculus. *Research in Engineering Design*, 1(3/4):187–203, 1990.

[191] K. L. Wood, K. N. Otto, and E. K. Antonsson. A Formal Method for Representing Uncertainties in Engineering Design. In P. Fitzhorn, editor, *First International Workshop on Formal Methods in Engineering Design*, pages 202–246, Fort Collins, Colorado, January 1990. Colorado State University.

[192] K. L. Wood, K. N. Otto, and E. K. Antonsson. Engineering Design Calculations with Fuzzy Parameters. *Fuzzy Sets and Systems*, 1992. Also appears in *Fuzzy Engineering toward Human Friendly Systems: Proceedings of the International Fuzzy Engineering Symposium '91*, Volume 1, 1991, T. Terano et. al., eds., IFES, LIFE, pages 434–445, Yokohama Japan.

[193] T. Woodson. *Introduction to Engineering Design.* McGraw-Hill, New York, 1966.

[194] R. Yager. Decision making with fuzzy sets. *Decision Sciences*, 6(3):590–600, 1975.

[195] R. Yager. Fuzzy decision making including unequal objectives. *Fuzzy Sets and Systems*, 1:87–95, 1978.

[196] R. Yager. Possibilistic decisionmaking. *IEEE Trans. on Systems Science and Cybernetics*, 9(7):388–392, July 1979.

[197] H. Yoshikawa. General design theory and a CAD system. In T. Sata and E. Warman, editors, *Man-Machine Communication in CAD/CAM*, pages 35–58. IFIP, North-Holland, 1981.

[198] H. Yoshikawa. CAD framework guided by general design theory. In K. Bø and F. Lillehagen, editors, *Proceedings of the IFIP WG5.2 Working Conference on CAD Systems Framework*, pages 241–256. IFIP, North-Holland, 1983.

[199] H. Yoshikawa. General design theory and artificial intelligence. In T. Bernold, editor, *Artificial Intelligence in Manufacturing: Key to Integration?*, pages 35–61. GDI, Elsevier Science Publishers, 1987.

[200] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[201] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning – I. *Information Sciences*, 8:199–249, 1975.

[202] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[203] L. A. Zadeh. Fuzzy sets versus probability. *Proc. IEEE*, 68(3):421, 1980.

[204] H. J. Zimmermann. *Fuzzy Set Theory - and Its Applications.* Management Science/Operations Research. Kluwer-Nijhoff Publishing, Boston, MA, 1985.