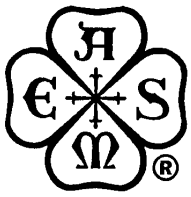
Proceedings of
The 1996 ASME Design Engineering Technical Conference and
Computers in Engineering Conference
August 18-22, 1996, Irvine, California



96-DETC/DTM-1524

MULTI-DIMENSIONAL MAPPING OF DESIGN IMPRECISION

William S. Law
Engineering Design Research Laboratory
California Institute of Technology
Pasadena, California 91125
law@design.caltech.edu

Erik K. Antonsson[1]
Engineering Design Research Laboratory
California Institute of Technology
Pasadena, California 91125
erik@design.caltech.edu

ABSTRACT

Preliminary design information is characteristically *imprecise* or fuzzy: specifications and requirements are subject to change, and the design description is vague and incomplete. The *Method of Imprecision* uses the mathematics of fuzzy sets to explicitly represent and manipulate imprecise preliminary design information, enabling the designer to better understand the full range of designs and performances that satisfy an imprecise set of specifications and requirements. This paper discusses the foundations of this methodology, and introduces pragmatic extensions that provide computationally tractable methods to map design imprecision from multiple design variables onto multiple performance variables. These methods attempt to minimize the number of function evaluations required while retaining an appropriate level of accuracy. This is achieved by using optimization to obtain extremal points for each performance variable and selectively applying a linear approximation for the mapping from the design variable space (DVS) to the performance variable space (PVS) to interpolate between extremal points. This linear approximation is constructed using regression techniques adapted from experiment design.

INTRODUCTION

Preliminary design information is characteristically *imprecise* or fuzzy: specifications and requirements are subject to change, and the design description is vague and incomplete. Precise information about the final design is usually not available. Yet many powerful evaluation tools, including finite element models, expect precisely specified data. Thus it is common for engineers to evaluate promising designs one by one. Alternatively, optimization may be used to search for the single "best" design. But these approaches focus on individual, precisely specified points in the design space and provide limited information about the full range of possible designs under consideration. An alternative approach would be to evaluate *sets* of designs and hence explicitly model design imprecision.

The *Method of Imprecision* (Wood, Otto, and Antonsson, 1992) uses the mathematics of fuzzy sets to represent and manipulate imprecise preliminary design information. This paper discusses the foundations of this methodology, and introduces pragmatic extensions that provide computationally tractable methods to map design imprecision from multiple design variables onto multiple performance variables.

THE METHOD OF IMPRECISION

Design alternatives are distinguished by the values of *design variables* $d_1, .., d_n$. Design variables do not completely specify a design: they serve only to distinguish alternatives that the designer considers to be distinct for the purpose of analysis. Hence if the designer is considering different lengths of a particular component, then that length should be a design variable. Other variables, such as the width of the same component, do not need to be design variables if they are not under active consideration.

The design variables $d_1, ..., d_n$ should be independent: no d_i should be a function of the other $d_1, ..., d_{i-1}, d_{i+1}, ..., d_n$. This does not imply that variables

[1] corresponding author

cannot be related to each other in any way, but merely that no variable be redundant. For example, two design variables *inner diameter* d_1 and *outer diameter* d_2 are clearly related in that $d_1 < d_2$, but neither is a function of the other. Tube thickness d_3, however, is a function of d_1 and d_2 and should not be defined as a third design variable. Note that in this particular example, it would probably be more convenient to choose as design variables *tube thickness* and then either *inner diameter* or *outer diameter*, to avoid having to ensure that *inner diameter* was less than *outer diameter*. The choice of design variables is thus not unique. The set of valid values for d_i is denoted $\mathcal{X}_i$. The whole set of design variables forms an n vector, $\vec{d}$, that distinguishes a particular design alternative in the *design variable space* (*DVS*).

In order to eliminate inferior design alternatives and refine the set of designs under consideration, designs need to be evaluated. Design evaluation seeks to predict how well a design alternative will perform when it is ultimately evaluated by the customer. "Performance", in general, has many aspects: rarely is it feasible to quantify all of them.

Performance variables $p_1, ..., p_q$ are the aspects of a design's performance that are explicitly quantified. Each performance variable p_j is defined by a mapping f_j such that $p_j = f_j(\vec{d})$. The mappings f_j can be any calculation or procedure to evaluate the performance of a design. A design variable can also be a measure of design performance and hence a performance variable. Weight, for example, could be a design variable describing distinct design alternatives while also being a performance variable that the customer is interested in. The set of valid values for p_j is denoted $\mathcal{Y}_j$. The set of performance variables for each design alternative forms a q vector, $\vec{p} = \vec{f}(\vec{d})$, that specifies the quantified performance of a design $\vec{d}$. Other aspects of performance which are not quantified are not formally modeled as performance variables, and are excluded from $\vec{p}$. The *performance variable space* (*PVS*) encompasses all quantified performances $\vec{p} = \vec{f}(\vec{d})$ that are achievable by designs $\vec{d} \in$ DVS.

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

Performance variables attempt to predict how the design will perform in the eyes of the customer, and hence for performance variables it is the customer's preferences that should be quantified, even if it is the designer who estimates them. Thus preferences on the time from 0 to 60 mph for an automobile, for example, are the customer's anticipated preferences, even if the customer has no stated preference on the 0–60 mph time in particular: the 0–60 mph time correlates with the average acceleration over the speed range of the engine (Michelena and Papalambros, 1995), which in turn correlates with perceived acceleration and vehicle performance. These issues, which together determine preferences on performance variables, will be referred to as *performance considerations*.

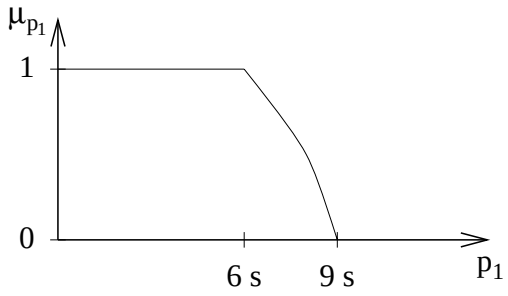


Figure 1. Imprecise functional requirement "less than 8 seconds".

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

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

$\mu_{p_j}(p_j)$ quantifies the customer's preference for values of p_j and is distinct from the customary membership function in a fuzzy set, which quantifies the extent to which values belong to the set. An example functional requirement on the performance variable *0–60 time* might at first be given as "less than 8 seconds". Further elicitation would reveal the imprecision associated with the nominally crisp value "8 seconds", resulting in a preference function μ_{p_1}, where p_1 is *0–60 time*, as shown in Figure 1. Values of $p_1 \leq 6$s have $\mu_{p_1} = 1$ and are most preferred or ideal. Values of $p_1 \geq 9$s have $\mu_{p_1} = 0$ and are unacceptable.

The basis for preferences among values of a design variable is less obvious. Design variables distinguish alternative designs: two different values of a design variable d_i indicate two distinct designs, but the values of d_i do not indicate if one design is preferred over the other. Design variables do not directly measure design performance and hence the customer has no direct basis for preferring any particular design variable value. The length of the rear axle of an automobile, for example, is not a variable of much interest to the typical customer. Yet choosing different axle lengths will affect aspects of design performance such as vehicle handling that *are* of interest to the customer. Some of these aspects will be quantified and hence modeled as performance variables. The customer's preferences on these performance consider-

ations are already represented as functional requirements. Other aspects of design performance that are not directly related to explicitly quantified performance variables are as yet not modeled, and the preferences that exist on these attributes have not been represented. These preferences, arising from aspects of design performance that are not represented by performance variables, are represented as preferences on the design variables $d_1, ..., d_n$.

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

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

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

1. to determine which unquantified aspects of performance to consider and their relative importance,
2. to estimate how values of the design variable d_i affect each unquantified aspect of performance considered, and
3. to anticipate the customer's preferences on these unquantified aspects of performance.

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

In order to evaluate designs $\vec{d} \in \text{DVS}$, the various individual preferences must be combined or aggregated to give a single, overall measure. This aggregation, in practice, occurs in two stages. First, the individual design preferences μ_{d_i} are aggregated into the combined design preference μ_d and the individual functional requirements μ_{p_j} are aggregated into the combined functional requirement μ_p. Second, μ_d and μ_p are aggregated into the *overall preference* μ_o, which combines all of the preferences specified:

$$\begin{aligned}\mu_o &= \mathcal{P}_{III}\left(\mathcal{P}_I(\mu_{d_1}, ..., \mu_{d_n}), \mathcal{P}_{II}(\mu_{p_1}, ..., \mu_{p_q})\right) \\ &= \mathcal{P}_{III}(\mu_d, \mu_p). \end{aligned} \quad (1)$$

The *aggregation functions* $\mathcal{P}_I$, $\mathcal{P}_{II}$, and $\mathcal{P}_{III}$ reflect the trade-off strategies for each aggregation, *i.e.*, how competing attributes of the design should be traded-off against each other (Otto and Antonsson, 1991; Scott and Antonsson, 1996). The trade-off strategy formalizes the designer's balancing of conflicting goals and constraints. The choice of aggregation function is not one that the designer is free to make: the appropriate trade-off strategy is usually dictated by the design problem. Although it is the designer who trades-off the different attributes of the design, it is the relationship between attributes, a property of the design problem itself, that determines how they should be traded-off.

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

$$\mu_o(\vec{d}) = \min\left(\mu_{d_1}, ..., \mu_{d_n}, \mu_{p_1}, ..., \mu_{p_q}\right). \quad (2)$$

This is Bellman and Zadeh's (1970) hard "and" operation for fuzzy sets, which does not allow attributes to be traded-off against each other. Yager (1978) notes that this choice of aggregation function leads to the classic max-min solution from game theory.

Alternatively, consider an ordinary household battery, and in particular the performance variables battery life (energy stored) and unit cost. A different relationship exists between these two variables. Low unit cost can partially compensate for short battery life and long battery life can partially compensate for high unit cost. The two attributes unit cost and battery life can be traded-off against each other, so that a more acceptable attribute partially compensates for a less acceptable attribute. This can be modeled as a *fully compensating* trade-off strategy for which the aggregation function is the geometric weighted mean or product

of powers $\mathcal{P}_\Pi$:

$$\mu_o(\vec{d}) = \left(\prod_{i=1}^{n} \mu_{d_i} \prod_{j=1}^{q} \mu_{p_j} \right)^{\frac{1}{n+q}} . \quad (3)$$

This is Bellman and Zadeh's (1970) soft "and" operation for fuzzy sets, which corresponds to the Nash solution from game theory (Yager, 1978).

Within a single design problem, different groups of attributes may require different trade-off strategies. In general, preferences for individual attributes will need to be successively aggregated by a hierarchy of different trade-off strategies.

After specifying design preferences $\mu_{d_1}(d_1), ..., \mu_{d_n}(d_n)$ and functional requirements $\mu_{p_1}(p_1), ..., \mu_{p_q}(p_q)$, and identifying the appropriate hierarchy of trade-off strategies, the individual $\mu_{d_i}(d_i)$ are aggregated to obtain $\mu_d(\vec{d})$, the combined design preference on the DVS. $\mu_d(\vec{d})$ is then mapped onto the PVS, using the extension principle (Zadeh, 1965):

$$\mu_d(\vec{p}) = \sup\{\mu_d(\vec{d}) \mid \vec{p} = \vec{f}(\vec{d})\} \quad (4)$$

where sup over the null set is defined to be zero. $\mu_d(\vec{p})$ is the combined design preference mapped onto the PVS.

Design preference functions are discretized into M preference values $\alpha_1, ..., \alpha_M$ so that mapping μ_d from the DVS to the PVS is computationally tractable. Individual level intervals $[d_{i\,\min}^{\alpha_k}, d_{i\,\max}^{\alpha_k}]$ in each d_i are combined into α-cut sets $D_{\alpha_1}, ..., D_{\alpha_M}$ in the DVS:

$$D_{\alpha_k} = \{\vec{d} \in \text{DVS} \mid \mu_d(\vec{d}) \geq \alpha_k\} \quad (5)$$

where $k = 1, ..., M$. The combined design preference on the DVS, $\mu_d(\vec{d})$, is described by $D_{\alpha_1}, ..., D_{\alpha_M}$. Note that D_ϵ (the α-cut at infinitesimal $\alpha = \epsilon$, $0 < \epsilon \ll 1$) contains all α-cuts with $\alpha > \epsilon$ and includes all the design alternatives with at least minimally acceptable (non-zero) design preference. D_ϵ defines the set of designs that are acceptable with respect to design preferences, *i.e.*, with respect to design considerations: the unquantified aspects of design performance not represented by performance variables. This specifically excludes preferences arising from performance considerations: the quantified aspects of performance represented by performance variables.

The combined design preference mapped onto the PVS is described by α-cut sets $P_{\alpha_1}, ..., P_{\alpha_M}$:

$$P_{\alpha_k} = \{\vec{p} \in \text{PVS} \mid \mu_d(\vec{p}) \geq \alpha_k\} \quad (6)$$

where $k = 1, ..., M$.

The following is a summary of the notation introduced:

d_i	design variable, $i = 1, ..., n$
p_j	performance variable $p_j = f_j(\vec{d})$, $j = 1, ..., n$
$\mu_{d_i}(d_i)$	individual design preference for d_i, $i = 1, ..., n$
$\mu_d(\vec{d})$	combined design preference on the DVS
$\mu_d(\vec{p})$	combined design preference mapped onto the PVS
$\mu_{p_j}(p_j)$	individual functional requirement for p_j, $j = 1, ..., q$
$\mu_p(\vec{p})$	combined functional requirement on the PVS
$\mu_o(\vec{d})$	overall preference on the DVS
$\mu_o(\vec{p})$	overall preference on the PVS
$\mathcal{P}_{\min}$	non-compensating aggregation function
$\mathcal{P}_\Pi$	fully compensating aggregation function
D_{α_k}	α-cut in the DVS, within which $\mu_d(\vec{d}) \geq \alpha_k$, $k = 1, ..., M$
P_{α_k}	α-cut in the PVS, within which $\mu_d(\vec{p}) \geq \alpha_k$, $k = 1, ..., M$

MAPPING DESIGN IMPRECISION

In implementing the Method of Imprecision, a key step is mapping design preference μ_d from the n-dimensional design variable space (DVS) to the q-dimensional performance variable space (PVS). The design preference α-cuts D_{α_k} and P_{α_k} only define a convenient discretization of the combined design preference function $\mu_d(\vec{d})$. What is needed is an algorithm to map the D_{α_k} to the P_{α_k}. Such an algorithm should be both efficient in terms of function evaluations and sufficiently accurate.

If the individual design preferences are to be combined with a non-compensating aggregation function $\mathcal{P}_{\min}$, the individual design preference level intervals give rise to a combined design preference that is fully described by α-cut sets $D_{\alpha_1}, ..., D_{\alpha_M}$ that are n-cubes in the DVS.

$$D_{\alpha_k} = [d_{1\,\min}^{\alpha_k}, d_{1\,\max}^{\alpha_k}] \times \ldots \times [d_{n\,\min}^{\alpha_k}, d_{n\,\max}^{\alpha_k}]. \quad (7)$$

For aggregation functions other than $\mathcal{P}_{\min}$, the $D_{\alpha_1}, ..., D_{\alpha_M}$ n-cubes do not fully describe the combined design preference $\mu_d(\vec{d})$. Additional, non-n-cubic level sets must be defined to correctly represent the aggregation of the discretized individual design preferences $\mu_{d_1}, ..., \mu_{d_n}$ into the combined design preference $\mu_d(\vec{d})$. Nevertheless, the methods developed below are easily extended to also map these non-n-cubic level sets. For the purposes of describing the method, it is assumed that the design preference level sets to be mapped are n-cubes.

Each D_{α_k} maps onto an α-cut P_{α_k} in the PVS via $\vec{f}$: DVS $\rightarrow$ PVS ($\vec{p} = \vec{f}(\vec{d})$). P_{α_k} will in general be neither a q-cube, nor even defined by straight edges. But given the need to minimize the number of function evaluations and the preliminary nature of the design information, the exact geometry of P_{α_k} need not be calculated: an approximation

is sufficient. Indeed, as q, the number of performance variables, increases beyond two, there is little reason to pursue more accurate results that may be difficult or impossible to interpret.

A straightforward extension of the Method of Imprecision to deal with multiple performance variables would use interval methods to obtain intervals $[{p_j}_{\min}^{\alpha_k}, {p_j}_{\max}^{\alpha_k}]$ for each p_j independently, and then combine these to obtain a q-cube approximation to P_{α_k}:

$$P_{\alpha_k}^{\square} = [{p_1}_{\min}^{\alpha_k}, {p_1}_{\max}^{\alpha_k}] \times, \ldots, \times [{p_q}_{\min}^{\alpha_k}, {p_q}_{\max}^{\alpha_k}]. \qquad (8)$$

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

A superior approach is to approximate $\vec{f}$ as some simple analytic function $\vec{f}'$ over D_ϵ (the α-cut at infinitesimal $\alpha = \epsilon$, which includes all the design alternatives under consideration) and hence map D_{α_k} directly onto the PVS, using $\vec{f}'$. A linear approximation is the obvious first choice:

$$\vec{f}'(\vec{d}) = \begin{bmatrix} f_1'(\vec{d}) \\ \vdots \\ f_q'(\vec{d}) \end{bmatrix} = \mathbf{A}(\vec{d} - \vec{d}_{\text{ctr}}) + \vec{\Delta}$$

$$= \begin{bmatrix} a_{11} \ldots a_{1n} \\ \vdots \quad \vdots \\ a_{q1} \ldots a_{qn} \end{bmatrix} \begin{bmatrix} d_1 - d_1^{\text{ctr}} \\ \vdots \\ d_n - d_n^{\text{ctr}} \end{bmatrix} + \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_q \end{bmatrix} \qquad (9)$$

where $\vec{d}_{\text{ctr}}$ is the center point of D_ϵ and Δ_j is the distance that f_j' is offset from f_j at $\vec{d}_{\text{ctr}}$. Although a linear approximation is not the only choice, higher order approximations introduce additional complexity, both in the shape of the level sets mapped onto the PVS and in the computational algorithm, that is not clearly justified.

A linear approximation will be adequate if $\vec{f}$ is not strongly non-linear and the precise geometry of P_{α_k} is not required. During preliminary design approximate answers are sufficient. Moreover, the difficulty of interpreting an irregular P_{α_k} set with curved boundaries in more than two dimensions suggests that a higher order approximation may be unnecessary for problems with more than two performance variables. Yet it is important to consider where on P_{α_k} more accurate results might be useful. Although the detailed geometry of P_{α_k} need not be known precisely, critical points on its boundary will be used to make design decisions, and these need to be determined with greater accuracy and reliability. The extremal points on the boundary of P_{α_k} in each of the p_j directions fall into this category. These points

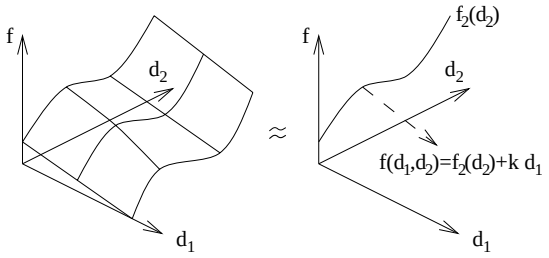


Figure 2. Reducing the Search Space with Linear Regression.

are determined by optimization in the design variables that are not acceptably linear (Law and Antonsson, 1995).

Obtaining a linear approximation $\vec{f}'$ fulfills two purposes: it removes acceptably linear design variables from the search space for optimization (Figure 2), and it provides a global approximation to $\vec{f}$ for determining the geometry of P_{α_k} between extremal points. The mapping of D_{α_k} onto the PVS does not, however, depend entirely upon the accuracy of the linear approximation $\vec{f}'$. The shape of P_{α_k} in the PVS is constructed by obtaining extremal points in each p_j via optimization (facilitated by linear approximation), and then interpolating the bounding edges between points using $\vec{f}'$. It is not expected that many performance variables will be well modeled by a linear approximation, even within a limited region of the DVS. But the linear approximation $\vec{f}'$ furnishes additional information about the shape of P_{α_k} away from extremal points that would otherwise be unavailable. $\vec{f}'$ is used to provide approximate information not to replace precise information, but to replace a lack of information. Useful information about P_{α_k} can still be obtained even where $f_1, \ldots, f_q$ are all highly non-linear. If $\vec{f}$ is non-linear, the calculated shape of P_{α_k} will be incorrect. But the extremal points obtained using optimization do not depend upon $\vec{f}$ being linear. Moreover, even if $\vec{f}'$ is completely unacceptable because $f_1, \ldots, f_q$ are all strongly non-linear in $d_1, \ldots, d_n$, $P_{\alpha_k}^{\square}$ defined by the extremal values in each p_j provides a bounding set for P_{α_k}.

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

$$\begin{bmatrix} a_{11} \\ \vdots \\ a_{q1} \end{bmatrix}, \ldots, \begin{bmatrix} a_{1n} \\ \vdots \\ a_{qn} \end{bmatrix}. \qquad (10)$$

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

FUNCTION APPROXIMATION

The linear approximations $f'_1, ..., f'_q$ are obtained using techniques adapted from design of experiments. These techniques rely on orthogonal arrays, which specify an efficient, independent set of points at which the function is evaluated. Orthogonal arrays are widely used not only for statistical design of experiments but also for the related Taguchi Method or Robust Design methodology (Peace, 1993; Phadke, 1989) and even their direct application to engineering design is not new. Chi and Bloebaum (1995) describe a simple and practical application of orthogonal arrays to a material selection problem for multi-bar trusses. Korngold and Gabriele (1995) use experiment design to construct a global quadratic approximation for a multidisciplinary problem: their methods are similar to those that have been adopted for Method of Imprecision calculations. A fundamental difference, however, is that Korngold and Gabriele have sought to solve a highly complex and general problem from a necessarily abstract and mathematical perspective. The techniques introduced in (Law and Antonsson, 1995) and extended in this paper adopt a pragmatic approach that attempts to address the concerns of potential users. Engineers at one major domestic automobile manufacturer, for example, consider each function evaluation to be a significant cost: it takes 60 seconds on a supercomputer to evaluate a *simplified* finite element model of a vehicle structure. Furthermore, most engineers do not have the time to become experts on statistical techniques. New methodologies are expected to come pre-packaged as out-of-the-box software. Using experiment design to obtain linear regression models is efficient in function evaluations, does not require advanced statistical techniques, and is well suited to computer implementation.

In (Law and Antonsson, 1995) the use of a fractional factorial experiment to obtain f'_j was discussed. This section will describe further techniques for identifying unacceptably non-linear design variables, and for determining the constant offsets $\Delta_1, ..., \Delta_q$ in Equation 9.

A fractional factorial experiment evaluates a balanced subset of the 2^n corner points of the D_ϵ n-cube. The center point of D_ϵ is also evaluated, and is used to estimate

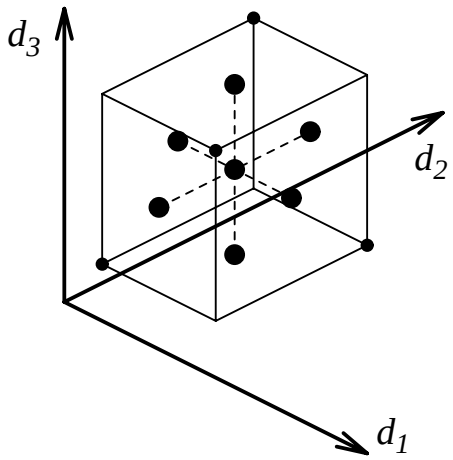


Figure 3. Central Composite Design.

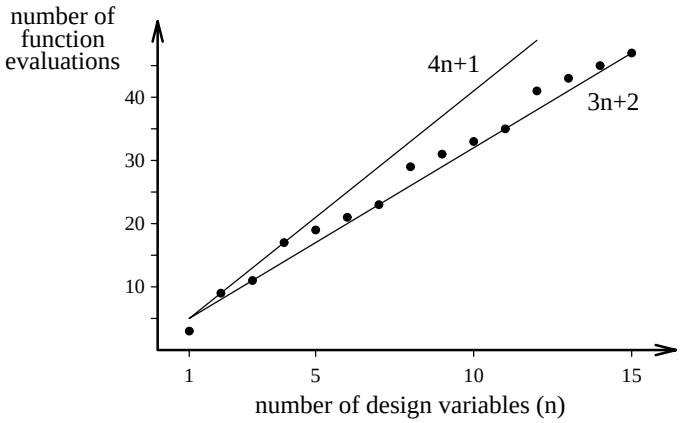


Figure 4. Number of Function Evaluations Required.

the non-linearity and non-monotonicity of f_j. Since the fractional factorial only evaluates corner points, comparison with the centerpoint can only indicate whether f_j is non-monotonic and the degree to which it is non-linear, and cannot distinguish the design variable that f_j is non-monotonic or non-linear in. If f_j is non-linear in d_i, f'_j will be inaccurate in d_i, but if f_j is non-monotonic in d_i, f'_j will be invalid in d_i. Monotonicity in d_i is the minimum condition for d_i to be acceptably linear. In order to estimate non-linearity and non-monotonicity of f_j in each d_i, an additional *one-factor-at-a-time* experiment is conducted. Figure 3 shows the points that would be evaluated for a resolution III fractional factorial experiment (small dots) and a one-factor-at-a-time experiment (large dots) where there are three design variables. The combined experiment is termed a *(face-centered) central composite design* (Montgomery, 1991).

The resolution of a fractional factorial experiment indicates the extent to which interactions are separated: a

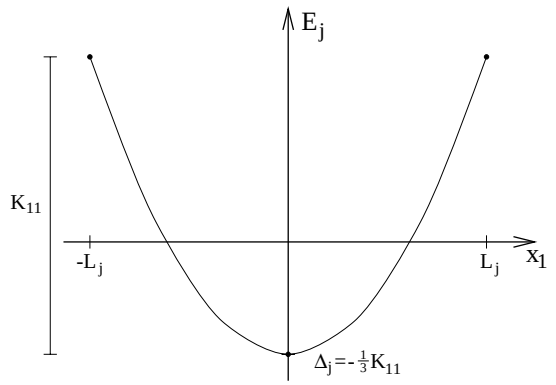


Figure 5. Minimizing $\int_{-L_j}^{L_j} {E_j}^2\, dx$ with offset Δ_j ($E_j = K_{11}{x_1}^2$).

resolution IV experiment provides more reliable information than a resolution III experiment but requires more function evaluations (Law and Antonsson, 1995). Since the one-factor-at-a-time experiment is an additional check for non-linearity and non-monotonicity, a resolution III experiment is sufficient. Between $3n+2$ and $4n+1$ (for $n>1$) function evaluations are required for such a central composite design (Figure 4).

After obtaining the linear regression matrix $\mathbf{A}$, the offsets $\Delta_1, ..., \Delta_q$ must be determined (see Equation (9)). $f_j'(\vec{d})$ must approximate $f_j(\vec{d})$ over the entire search space D_ϵ. Setting $\Delta_j = 0$ would give a Taylor approximation which is accurate near $\vec{d}_{\text{ctr}}$ only. Setting Δ_j such that $f_j'(\vec{d})$ passes through the mean value of $f_j(\vec{d})$ over all *evaluated* points would give a close approximation near the boundaries of D_ϵ only, since only one interior point $\vec{d}_{\text{ctr}}$ is evaluated. The Taylor expansion of $f_j(\vec{d})$ near $\vec{d}_{\text{ctr}}$ indicates that the residual error in approximating f_j is equal to the offset Δ_j plus second and higher order terms ($\vec{x} = \vec{d} - \vec{d}_{\text{ctr}}$ where $-L_j \le x_j \le L_j \;\; j = 1, ..., q$):

$$\begin{aligned} E_j(\vec{d}) &= f_j'(\vec{d}) - f_j(\vec{d}) \\ &\approx \Delta_j + K_{11}{d_1}^2 + K_{22}{d_2}^2 + ... \\ &\quad + K_{12}d_1d_2 + K_{13}d_1d_3 + ... \end{aligned} \tag{11}$$

For the purpose of determining an appropriate value for Δ_j, assume that the error in approximating f_j is predominantly quadratic. In order to minimize $\int_{D_\epsilon} {E_j}^2\, \text{dV}$ (the square error integrated over D_ϵ), Δ_j should be set to $-\frac{1}{3}\sum_{i=1}^n K_{ii}$ (Figure 5 illustrates a one-dimensional example). Qualitatively, it is clear that cross-terms ($K_{ik}x_ix_k$ where $i \neq k$) do not introduce the need to offset f_j' since they lead to an error that is anti-symmetric in x_i and x_k. Square terms ($K_{ii}{x_i}^2$), however, do require a constant offset to minimize $\int_{D_\epsilon} {E_j}^2\, dV$ since they introduce an error in one direction only (positive or negative). The errors (for $\Delta_j = 0$) at the $N^\square$ corner points $\vec{d}_1^{\,\square}, ..., \vec{d}_{N^\square}^{\,\square}$ evaluated by the fractional factorial experiment nominally average to $\sum_{i=1}^n K_{ii}$ (cross-terms cancel). The error at each point $\vec{d}_i^{\,-}$ and $\vec{d}_i^{\,+}$ evaluated in the d_i direction for the one-factor-at-a-time experiment is nominally equal to K_{ii} (cross-terms equal zero). The sum of these errors is used to estimate $-\frac{1}{3}\sum_{i=1}^n K_{ii}$:

$$\begin{aligned} \Delta_j &= -\frac{1}{3(N^\square + 2)} \left(\sum_{h=1}^{N^\square} E_j(\vec{d}_h^{\,\square} - \vec{d}_{\text{ctr}}) \right. \\ &\qquad \left. + \sum_{i=1}^{n} \left(E_j(\vec{d}_i^{\,-} - \vec{d}_{\text{ctr}}) + E_j(\vec{d}_i^{\,+} - \vec{d}_{\text{ctr}}) \right) \right) \\ &= -\frac{1}{3(N^\square + 2)} \left(\sum_{h=1}^{N^\square} f_j(\vec{d}_h^{\,\square}) + \sum_{i=1}^{n} \left(f_j(\vec{d}_i^{\,-}) + f_j(\vec{d}_i^{\,+}) \right) \right. \\ &\qquad \left. - (N^\square + 2n) f_j(\vec{d}_{\text{ctr}}) \right) \end{aligned} \tag{12}$$

where $E_j(\vec{x})$ is calculated for $\Delta_j = 0$. The points $\vec{d}_1^{\,\square}, ..., \vec{d}_{N^\square}^{\,\square}$ and $\vec{d}_1^{\,\pm}, ..., \vec{d}_n^{\,\pm}$ are symmetric on D_ϵ such that the linear approximation f_j' averages to $f_j(\vec{d}_{\text{ctr}})$ on D_ϵ (for $\Delta_j = 0$). Separating $f_j(\vec{d}_{\text{ctr}})$ in Equation (12) simplifies the calculation.

As unacceptably non-linear design variables are eliminated from f_j', Δ_j must be re-calculated since the square error in the design variable that was eliminated no longer needs to be offset. f_j will be calculated in all eliminated, non-linear directions and approximated only in the remaining, linear directions (Figure 2). The re-calculation of Δ_j is complicated by the need to balance the data such that the result is not overly dependent on any single evaluated point. Every non-linear design variable that is eliminated reduces the space that f_j' approximates by one dimension. The points $\vec{d}_i^{\,-}$ and $\vec{d}_i^{\,+}$ evaluated in every remaining, linear direction (d_i such that $i \in \mathcal{L}$) lie within this reduced space. But all of the fractional factorial corner points $\vec{d}_1^{\,\square}, ..., \vec{d}_{N^\square}^{\,\square}$ lie outside the reduced space. These corner points now provide less relevant information that must be adjusted: the K_{ii} such that $i \notin \mathcal{L}$, corresponding to eliminated design variables, must be subtracted from the estimate for $\sum_{i=1}^n K_{ii}$ obtained from the error at each corner point. Moreover, each K_{ii} $i \notin \mathcal{L}$ can only be estimated from the errors at two evaluated points: $\vec{d}_i^{\,-}$ and $\vec{d}_i^{\,+}$. Hence if f_j is not acceptably linear in all n design variables, a different formula must be used to estimate Δ_j:

$$\Delta_j = -\frac{1}{3(2 + n_{\mathcal{L}})} \left(\frac{2}{N^\square} \sum_{h=1}^{N^\square} f_j(\vec{d}_h^{\,\square}) - \sum_{i \notin \mathcal{L}} \left(f_j(\vec{d}_i^{\,-}) + f_j(\vec{d}_i^{\,+}) \right) \right.$$

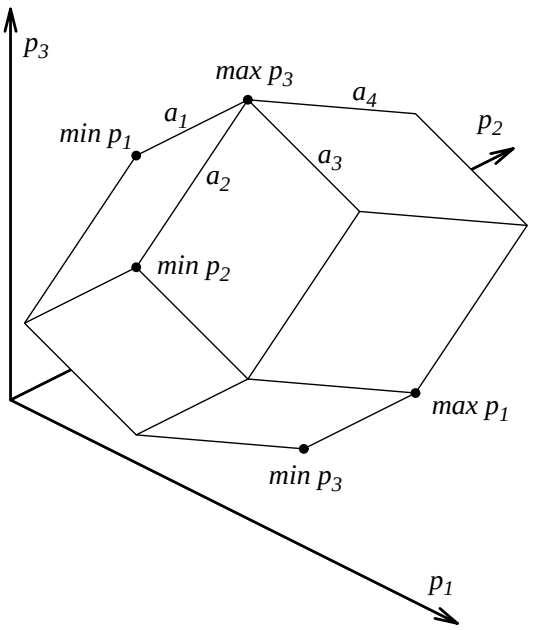


Figure 6. Approximated α-cut P'_ϵ on the PVS.

$$+\sum_{i\in\mathcal{L}}\left(f_j(\vec{d}_i^{-})+f_j(\vec{d}_i^{+})\right)-2n_{\mathcal{L}}\,f_j(\vec{d}_{\text{ctr}})\Bigg) \qquad (13)$$

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

A special case arises when there is only one remaining linear design variable for f_j, *i.e.*, $n_{\mathcal{L}}=1$. Since f'_j will only be used to approximate f_j in a single design variable $d_{i\in\mathcal{L}}$, Δ_j can be directly estimated from $\vec{d}^{-}_{i\in\mathcal{L}}$ and $\vec{d}^{+}_{i\in\mathcal{L}}$ only, as in Figure 5:

$$\Delta_j=-\frac{1}{6}\left(f_j(\vec{d}^{-}_{i\in\mathcal{L}})+f_j(\vec{d}^{+}_{i\in\mathcal{L}})-2f_j(\vec{d}_{\text{ctr}})\right). \qquad (14)$$

EXAMPLE

Suppose that for a particular design problem, there are $n=4$ design variables and $q=3$ performance variables. The designer wishes to conduct a preliminary analysis to explore how preferences on the design and performance variables intersect. Individual design preferences $\mu_{d_1},\mu_{d_2},\mu_{d_3},\mu_{d_4}$ are specified as intervals $[d_{i\min}^{\alpha_k},d_{i\max}^{\alpha_k}]$ at two preference levels: $\alpha_1=\epsilon,\alpha_2=1$. An infinitesimal yet non-zero preference ϵ indicates a barely acceptable variable value. The α-cut interval at $\alpha_1=\epsilon$ identifies the largest acceptable range of values for the variable. Values outside this range have zero preference and are thus unacceptable. Conversely, a preference of one indicates an ideal variable value. The α-cut interval at $\alpha_2=1$ identifies the ideal or target range of values for the variable. Specifying only two α-cuts is a minimal implementation that is limited to calculating the boundaries of two sets: the set of acceptable designs and the set of ideal designs. Relatively few function evaluations are required. This is consistent with a preliminary analysis. For this minimal implementation, the distinction between aggregation functions is eliminated: at this level of discretization, the design preference intervals lead to the same n-cubic combined design preference α-cuts D_ϵ and D_1 (see Equation (7)) regardless of the aggregation functions used. These n-cubic α-cuts fully describe the combined design preference μ_d on the DVS. A linear approximation to the mapping $\vec{f}$: DVS $\rightarrow$ PVS is obtained by evaluating a central composite design over D_ϵ and constructing the matrix $\mathbf{A}$:

$$\mathbf{A}=\begin{bmatrix}1.1 & -0.12 & 2 & 0.68\\ -0.1 & 1 & 0.24 & 2\\ 1.1 & 0.88 & -1.24 & 1\end{bmatrix}. \qquad (15)$$

P'_ϵ, the projection of D_ϵ onto the PVS via $\mathbf{A}$, is shown in Figure 6. P'_ϵ is a convex polyhedron that approximates the actual α-cut P_ϵ. Four edges that correspond to the four columns of $\mathbf{A}$ are labeled $a_1,...,a_4$. These directions are the principal directions $d_1,...,d_4$ mapped onto the PVS. The labeled corners are extrema in p_j: these points would be obtained by optimization. The conventional optimization approach to this problem would be to search for the optimal p_1, p_2, or p_3, within a constrained search space such as D_ϵ. Thus if p_1 is to be maximized and p_2 and p_3 are to be minimized, the information provided by optimization would be limited to three of the points in Figure 6 labeled max p_1, min p_2, and min p_3. Instead, the Method of Imprecision uses optimization to find both extrema in each p_j and then constructs an approximation to the entire set P_ϵ using the linear approximation $\vec{f}'$. The accuracy of the extrema is dependent only on the accuracy of the optimization algorithm used. Linear approximation is used to provide additional information: to fill in the gaps between extrema and paint a more complete picture.

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

P'_1, which approximates the approximate region in the

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

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

with respect to the preferences specified on the design and performance variables. Additionally, the effect of functional requirements on design performances is represented by the variation of $\mu_o(\vec{p})$ between the extremes ϵ and one. Due to discretization, however, the combined design preference μ_d is only mapped at the two extremes, and hence intermediate values of $\mu_o(\vec{p})$ do not reflect the true variation of $\mu_d(\vec{p})$. The combined functional requirement $\mu_p(\vec{p})$ is mapped back onto the DVS using the linear approximation $\vec{f}'(\vec{d})$. The resulting functional requirement on the DVS, $\mu_p(\vec{d})$, is aggregated with the combined design preference $\mu_d(\vec{d})$ to obtain the overall preference $\mu_o(\vec{d})$. $\mu_o(\vec{d})$ identifies the ideal ($\mu_o(\vec{d}) = 1$) and acceptable ($\mu_o(\vec{d}) \geq \epsilon$) sets of designs described above.

CONCLUSION

The Method of Imprecision represents imprecision through the customer's preferences on relevant aspects of design performance.

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

These preferences are aggregated separately on the design variable space (DVS) and the performance variable space (PVS). A key step is to map design preferences from the DVS to the PVS, a step that requires the mapping of *sets* as opposed to individual points. The extended methods introduced in this paper map design preference level sets from n design variables to q performance variables. These methods attempt to minimize the number of function evaluations required while retaining an appropriate level of accuracy. This is achieved by using optimization to obtain extremal points for each performance variable and selectively applying a linear approximation for $\vec{f}$: DVS $\rightarrow$ PVS to interpolate between extremal points.

ACKNOWLEDGMENT

This material is based upon work supported, in part, by: the National Science Foundation under NSF Grant Numbers DDM-9201424 and DMI-9523232. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

Bellman, R. E., and Zadeh, L. A., 1970, "Decision-making in a Fuzzy Environment," *Management Science*, Vol. 17, (No. 4), pp. B–141–163.

Chi, H. W., and Bloebaum, C. L., 1995, "Mixed Variable Optimiation Using Taguchi's Orthogonal Arrays," In *Advances in Design Automation - 1995*, Vol. DE-82, pp. 501–508 New York. ASME.

Korngold, J. C., and Gabriele, G. A., 1995, "Multidisciplinary Analysis and Optimization of Discrete Problems Using Response Surface Methods," In *Advances in Design Automation - 1995*, Vol. DE-82, pp. 173–180 New York. ASME.

Law, W. S., and Antonsson, E. K., 1995, "Optimization Methods for Calculating Design Imprecision," In *Advances in Design Automation - 1995*, Vol. 1, pp. 471–476. ASME.

Michelena, N. F., and Papalambros, P. Y., 1995, "Optimal Model-Based Decomposition of Powertrain System Design," In *Advances in Design Automation - 1995*, Vol. DE-82, pp. 165–172 New York. ASME.

Montgomery, D. C., 1991, *Design and Analysis of Experiments* Wiley, New York.

Otto, K. N., and Antonsson, E. K., 1991, "Trade-Off Strategies in Engineering Design," *Research in Engineering Design*, Vol. 3, (No. 2), pp. 87–104.

Peace, G. S., 1993, *Taguchi Methods: a Hands-On Approach* Addison-Wesley, Reading, MA.

Phadke, M., 1989, *Quality Engineering Using Robust Design* Prentice Hall, Englewood Cliffs, NJ.

Scott, M. J., and Antonsson, E. K., 1996, "Aggregation Functions for Engineering Design Trade-offs," *Fuzzy Sets and Systems* Submitted for review, July, 1995.

Wood, K. L., Otto, K. N., and Antonsson, E. K., 1992, "Engineering Design Calculations with Fuzzy Parameters," *Fuzzy Sets and Systems*, Vol. 52, (No. 1), pp. 1–20.

Yager, R., 1978, "Fuzzy Decision Making Including Unequal Objectives," *Fuzzy Sets and Systems*, Vol. 1, pp. 87–95.

Zadeh, L. A., 1965, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338–353.