ON THE USE OF STATISTICS IN DESIGN AND THE IMPLICATIONS FOR DETERMINISTIC COMPUTER EXPERIMENTS

Timothy W. Simpson, Jesse D. Peplinski, Patrick N. Koch, and Janet K. Allen
Systems Realization Laboratory
Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0405, USA

ABSTRACT
Perhaps the most prevalent use of statistics in engineering design is through Taguchi's parameter and robust design - using orthogonal arrays to compute signal to noise ratios in a process of design improvement. In our view, however, there is an equally exciting use of statistics in design that could become just as prevalent: it is the concept of metamodeling, whereby statistical models are built to approximate detailed computer analysis codes. Although computers continue to get faster, our analysis codes always seem to keep pace, so that their computational time remains non-trivial. Through metamodeling, approximations of these codes are built that are orders of magnitude cheaper to run. These metamodels can then be linked to optimization routines for fast analysis, or they can serve as a bridge for integrating codes across different domains.

In this paper we first review metamodeling techniques that encompass the Design of Experiments, Response Surface Methodology, Taguchi methods, neural networks, inductive learning, and kriging. We discuss their existing applications in engineering design and then address the dangers of applying traditional statistical techniques to approximate deterministic computer analysis codes. We conclude with recommendations for the appropriate use of metamodeling techniques in given situations and how common pitfalls can be avoided.

KEYWORDS: Design of Experiments, Response Surface Methodology, Metamodeling, Kriging, Neural Networks, Rule Induction, D-Optimal, Taguchi Robust Design

1 INTRODUCTION
Much of today's engineering analysis work consists of running complex computer analysis codes -- supplying a vector of design variables (inputs) \( \mathbf{x} \) and receiving a vector of responses (outputs) \( \mathbf{y} \). Despite continual advances in computing power and speed, the expense of running many analysis codes remains non-trivial. Single evaluations of aerodynamic or finite-element codes can take from minutes to hours, if not longer. What's more, this mode of query-and-response can often lead to a trial and error approach to design, where a designer may never uncover the functional relationship between \( \mathbf{x} \) and \( \mathbf{y} \) and therefore may never identify the best settings for input values.

Statistical techniques are widely used in engineering design to address these concerns. The basic approach is to construct approximations of the analysis codes that are much more efficient to run and that yield insight into the functional relationship between \( \mathbf{x} \) and \( \mathbf{y} \). If the true nature of a computer analysis code is represented as

\[
y = f(\mathbf{x}),
\]

then a "model of the model" or metamodel (Kleijnen, 1987) of the analysis code is taken to be

\[
\hat{y} = g(\mathbf{x}), \text{ and so } \mathbf{y} = \hat{y} + \epsilon
\]

where \( \epsilon \) represents both the error of approximation and measurement or random errors. The most common metamodeling approach is to apply the Design of Experiments (DOE) to identify an efficient set of computer runs \( (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) \) and then employ regression analysis to create a polynomial approximation of the computer analysis code. These approximations then can replace the existing code while providing:

- a better understanding of the relationship between \( \mathbf{x} \) and \( \mathbf{y} \) (we typically lose this understanding when running computer codes "blindly"),
- facilitated integration of domain dependent computer codes (we no longer work with the codes themselves but rather simple approximations of them), and
- fast analysis tools for optimization and exploration of the design space (we become more efficient because we are using the approximations rather than the computationally expensive codes themselves).

However, we have found that many published applications (including our own) utilizing these statistical methods for
computer-oriented design are statistically questionable, if not incorrect. The root cause is that many of our analysis codes are deterministic, which means that the error of approximation is not due to random effects. This calls into question most subsequent statistical analyses of model significance. Consequently, our purpose in this paper is to highlight the potential statistical pitfalls in metamodeling and to provide some general recommendations for the proper use of statistical techniques in design. In Section 2, we present a review of metamodeling techniques that encompasses regression, neural networks, inductive learning, and the more advanced approach of kriging. We conclude Section 2 with an introduction to the general statistical approaches of Response Surface Methodology (RSM) and Taguchi’s robust design. In Section 3, we describe the engineering design context for statistical applications, review existing applications, methods and tools, and close by taking a closer look at metamodeling techniques for deterministic computer experiments. In Section 4 we present some general recommendations for avoiding some important pitfalls in the application of metamodeling to computer-oriented design, and in Section 5 we conclude by discussing briefly some of the more advanced issues that continue to make metamodeling an active research area.

2 REVIEW OF METAMODELING TECHNIQUES

In its most basic sense, metamodeling involves (a) choosing an experimental design for generating data, (b) choosing a model to represent the data, and then (c) fitting the model to the observed data. There are a variety of options for each of these steps as shown in Figure 1, and we have attempted to highlight a few of the more prevalent ones. For example, Response Surface Methodology usually employs central composite designs, second order polynomials and least squares regression analysis.

<table>
<thead>
<tr>
<th>EXPERIMENTAL DESIGN</th>
<th>MODEL CHOICE</th>
<th>MODEL FITTING</th>
<th>SAMPLE METAMODERNING TECHNIQUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Fractional) Factorial</td>
<td>Polynomial</td>
<td>Least Squares Regression</td>
<td>Response Surface Methodology</td>
</tr>
<tr>
<td>Central Composite Design</td>
<td>Linear, quadratic, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Box-Behnken</td>
<td>Splines (linear, cubic)</td>
<td>Weighted Least Squares</td>
<td></td>
</tr>
<tr>
<td>D-Optimal</td>
<td>Frequency-Domain</td>
<td>Squares Regression</td>
<td></td>
</tr>
<tr>
<td>G-Optimal</td>
<td>Kernel Smoothing</td>
<td>Best Linear Unbiased</td>
<td></td>
</tr>
<tr>
<td>Orthogonal Array</td>
<td>Radial Basis Functions</td>
<td>Predictor (BLUP)</td>
<td></td>
</tr>
<tr>
<td>Plackett-Burman</td>
<td>Radial Basis or splines</td>
<td>Best Linear Predictor</td>
<td></td>
</tr>
<tr>
<td>Hexagon</td>
<td></td>
<td>(BLP)</td>
<td></td>
</tr>
<tr>
<td>Hybrid</td>
<td>Rulebase or Decision Tree</td>
<td>Log-Likelihood</td>
<td></td>
</tr>
<tr>
<td>Latin hypercube</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select By Hand</td>
<td>Rulebase or Decision Tree</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Selection</td>
<td>Weighted Least Squares</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spline</td>
<td>Square Regression</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1. Techniques for Metamodelling

In the remainder of this section we provide a brief overview of several of the options listed in Figure 1. In Section 2.1, we focus on experimental designs with particular emphasis on (fractional) factorial designs, central composite designs, and orthogonal arrays. We also discuss measures of merit for the evaluation of experimental designs. In Section 2.2, we discuss model choice and model fitting, focusing on response surfaces, neural networks, inductive learning and kriging. We conclude with an overview of two of the more prevalent metamodeling techniques, namely, Response Surface Methodology and Taguchi’s robust design.

2.1 Experimental Design

Properly designed experiments are essential for effective computer utilization. The traditional approach in engineering is to vary one parameter at a time within a computer analysis code and observe the effects, or to randomly assign different combinations of factor settings to be used as alternative analyses for comparison. Experimental design techniques developed for effective physical experiments, however, are being applied for the design of engineering computer experiments to increase the efficiency (computer time) and effectiveness of these analysis codes. In this section we present an overview of different types of experimental designs provided, along with measures of merit for selecting or comparing different designs.

2.1.1 A Survey of Experimental Designs. An experimental design formally represents a sequence of experiments to be performed, expressed in terms of factors (design variables) set at specified levels (predefined values). An experimental design is represented mathematically by a matrix X where the rows denote experimental runs and the columns denote the particular factor setting for each factor for each run.

Factorial Designs: The most basic experimental design is a full factorial design. The number of design points dictated by a full factorial design is the product of the number of levels for each factor. The most common designs are the $2^k$ (for evaluating main effects and interactions) and $3^k$ designs (for evaluating main and quadratic effects and interactions) for $k$ factors at 2 and 3 levels, respectively. A $2^3$ full factorial design is shown graphically in Figure 2(a).

The size of a full factorial experiment increases exponentially with the number of factors which may lead to an unmanageable number of experiments, e.g., 10 factors each with 2 levels requires $2^{10}$ or 1024 experiments. Fractional factorial designs can be used when experiments are costly and the number of design points for a full factorial design is large. A fractional factorial design consists of a fraction of a full factorial design; the most common fractional factorial designs are $2^{k-p}$ designs where a fraction is equal to $1/2^p$. A half fraction of the $2^3$ full factorial design is shown in Figure 2(b).

The reduction of the number of design points associated with a fractional factorial design does not come without a price. Design resolution and aliasing of effects are two critical (and related) issues which are important when selecting a fractional factorial design. The $2^3$ full factorial design shown in Figure 2(a) allows estimation of all main effects ($x_1$, $x_2$, $x_3$), all two factor interactions ($x_1x_2$, $x_1x_3$ and $x_2x_3$), as well as the three
factor interaction \((x_1x_2x_3)\). For the \(2^{3-1}\) fractional factorial indicated by the solid dots in Figure 2(b), the main effects are aliased (or biased) with the two factor interactions. Aliased effects cannot be estimated independently unless all but one of the aliased effects are known or assumed not to exist. The resolution of an experiment design is defined by the aliasing of effects: a design of resolution \(R\) is one in which no \(p\)-factor effect is aliased with any other effect containing less than \(R-p\) factors (Box, et al., 1978). The \(2^{3-1}\) design in Figure 2(b) can be at most a Resolution III design, denoted \(2^{3-1}_{III}\).

Often \(2^k\) and \(2^{k-p}\) designs are used for identifying or screening important factors when the number of factors is too large to evaluate higher order effects. In these cases the sparsity of effects principle (Montgomery, 1997) is invoked, in which the system is assumed to be dominated by the main effects and their low-order interactions. Based on this principle, two level fractional factorial designs can be used to "screen" the factors to determine which have the largest effect. The sparsity of effects principle may not always hold true, however. As noted by Hunter (1970), every design provides aliased estimates (quadratic and cubic effects, if present, will bias the estimates of the mean and main effects respectively when a two level factorial design is used). "Any phenomenon omitted from a fitted model will confound [i.e., alias] certain estimated parameters in the model regardless of the design used. Good fractional factorial designs are carefully arranged so that estimates of effects thought to be important are confounded by effects thought to be unimportant."

One specific family of fractional factorial designs commonly used for screening are the two level Plackett-Burman (P-B) designs. These resolution III designs are constructed to study \(k=N-1\) factors in \(N=4m\) design points. The class of P-B designs in which \(N\) is a power of two are called geometric designs and are identical to the usual \(2^{k-p}\) fractional factorials. When \(N\) is strictly a multiple of four, the P-B designs are referred to as non-geometric designs and have very messy alias structures. Their use in practical problems is often problematic particularly when the design is saturated, i.e., the number of factors is exactly \(N-1\). If all interactions are negligible, however, these designs allow unbiased estimation of all main effects, using only one more design point than the number of factors, with smallest possible variance (Box, et al., 1978). Minimum variance and minimum size designs will be discussed in Section 2.1.2. For a more complete discussion of factorial designs, confounding, and design resolution (see Myers and Montgomery, 1995).

![Figure 2. Basic Three-Factor Designs](image)

**Central Composite and Box-Behnken Designs**: For estimating quadratic effects the \(3^k\) or \(3^{k-p}\) designs can be used but often lead to an unmanageable number of design points, e.g., 7 factors at three levels requires \(3^7\) or 2187 experiments. The most common second-order designs, configured to reduce the number of design points, are central composite and Box-Beihnken designs.

A central composite design (CCD) is a two level \((2^k + 2^p\) factorial design, augmented by \(n_0\) center points and two "star" points positioned at \(\pm \alpha\) for each factor. This design, shown for three factors in Figure 2(c) consists of \(2^{k-p} + 2^p + n_0\) total design points to estimate \(2k + k(k-1)/2 + 1\) coefficients. Setting \(\alpha=1\) locates the star points on the centers of the faces of the cube (for three factors), giving a face-centered central composite (CCF) design. It should be noted that for values of \(\alpha\) other than 1, each factor is evaluated at five levels.

Often it is desirable to employ the smallest number of factors levels (e.g., three levels to fit a quadratic model) when creating an experimental design. One common, economical class of such designs is the Box-Beihnken designs. These designs are formed by combining \(2^k\) factorials with incomplete block designs (see Box and Draper, 1987). These designs do not contain any points at the vertices of the hypercube defined by the upper and lower limits for each factor. This is desirable if these extreme points are expensive or impossible to test. More information about central composite and Box-Beihnken design can be found in (Montgomery, 1997).

**Orthogonal Arrays**: The experiment designs used by Taguchi, called orthogonal arrays, are most often simply fractional factorial designs in two or three levels \((2^{k-p}\) and \(3^{k-p}\) designs). These arrays are constructed to reduce the number of design points necessary to evaluate the required effects; the two-level \(L_6\), \(L_{12}\), and \(L_{16}\) arrays, for example, allow 3, 11, and 15 factors/effects to be evaluated in 4, 12, and 16 design points respectively. In many cases, these designs are identical to those given by Plackett and Burman (Lucas, 1994). The definition of orthogonality, for these arrays and other experiment designs, is given in Section 2.1.2. An overview of Taguchi's approach to parameter design, within which the orthogonal arrays are implemented, is given in Section 2.3.

**2.1.2 Measures of Merit for Evaluating Experimental Designs.** Selecting the appropriate design is essential for effective experimentation. Experimenters must balance the desire to gain as much information as possible about the response-factor relationships with the cost of experimentation and need for efficiency (measured in number of runs). Several measures of merit are available and useful for evaluating and comparing experimental designs to ensure the appropriate experiment is designed.

**Orthogonality, Rotatability, Minimum Variance, and Minimum Bias**: Four desirable characteristics of an experimental design, to facilitate efficient estimates of the parameters, are orthogonality, rotatability, minimum variance, and minimum bias. A design is orthogonal if for every pair of factors \(x_i\) and \(x_j\) the sum of the crossproducts for the N design points is zero. For a first order model, the estimates of all coefficients will have minimum variance if the design is configured so that...
the variance of predictions \( \hat{y} \) will also have constant variance at a fixed distance from the center of the design, and the design will be rotatable.

For second order modeling, Hunter (1985) suggests that orthogonality loses its importance. "If the objective of the experimenter is to forecast a response at either present or future settings of \( x \), then an unbiased minimum variance estimate of the forecast \( \hat{y} \) is required. In the late 1950's Box and his co-workers demonstrated that rotatability....and the minimization of bias from higher order terms....were the essential criteria for good forecasting."

**Unsaturated/Saturated and Supersaturated Designs:** In many cases the primary concern in the design of an experiment is the size (number of runs) of the resulting design. Most experiment designs are unsaturated in that they contain at least two more design points than the number of factors. An experiment design is said to be saturated if the number of design points is equal to one more than the number of factor effects to be estimated. Saturated fractional factorial designs allow unbiased estimation of all main effects with smallest size and smallest possible variance (Box, et al., 1978). In a supersaturated design, then, the number of design points is less than or equal to the number of factors. The most common examples of saturated designs are the Plackett-Burman two level designs discussed in the previous section, and the related orthogonal arrays of Taguchi. Additional aspects of supersaturated designs can be found in (Draper and Lin, 1990a; Draper and Lin, 1990b).

For estimating second order effects, small composite designs have been developed to reduce the number of design points necessary for a composite design. A small composite design becomes saturated if the number of design points is \( 2k + k(k-1)/2 + 1 \) (the number of coefficients to be estimated for a full quadratic model). Myers and Montgomery (1995) note that recent work has suggested that these designs may not be good choices in all applications; additional discussion on small composite designs can be found in (Box and Draper, 1987; Lucas, 1974).

**D-Optimal, D-Efficiency, and G-Efficiency:** It is most desirable to use unsaturated designs, when building predictive models, unless running the necessary experiments is prohibitively expensive. For comparing experiments based on number of design points and information obtained, many measures of merit exist. A common measure is D-efficiency.

A design is said to be D-Optimal if for the the model the volume of the joint confidence region for the vector of unknown parameters is minimum. As a consequence, the global maximum of the determinant of \( X'X \) is achieved. Here \( X \) is the expanded design matrix, including \( N \) rows (one for each design point or run), and \( p \) columns (one for each coefficient to be estimated, including the overall mean). The D-efficiency statistic for comparing designs, given in Equation 1, compares the determinant value achieved for the design with the D-optimal determinant value, normalized by the size of the matrix for comparing different size designs.

\[
\text{D-efficiency} = \left( \frac{|X'X|}{|X'X|_{\text{D-optimum}}} \right)^{1/p}
\]  

The D-efficiency statistic is the primary criteria with which designs have been compared. Other statistics for comparing designs such as G-efficiency and A-optimality have also been formulated (see Myers and Montgomery, 1995). Having completed our review of experimental design alternatives and measures of merit, we can now turn to the issues of model choice and model fitting.

### 2.2 Model Choice and Model Fitting

After selecting an appropriate experimental design and performing the necessary computer runs, the next step involves choosing an approximating model and fitting method. Many alternative models and methods exist, but for practicality we review in this section the four that we have found to be most prevalent in the literature: response surfaces, neural networks, inductive learning and kriging.

#### 2.2.1 Response Surfaces

Given a response of interest, \( y \), and a vector of independent factors \( x \) thought to influence \( y \), the relationship between \( y \) and \( x \) can be written as follows:

\[
y = f(x) + \varepsilon,
\]

where \( \varepsilon \) represents random error which is assumed to be normally distributed with mean zero and standard deviation \( \sigma \). Since the true response surface function \( f(x) \) is usually unknown, a response surface \( g(x) \) is created to approximate \( f(x) \). Predicted values are then obtained using \( \hat{y} = g(x) \).

The most widely used response surface approximating functions are simple low-order polynomials. If little curvature appears to exist, the first-order polynomial given in Equation 4 can be employed. If significant curvature exists, the second-order polynomial in Equation 5, including all two-factor interactions, can be used.

\[
\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i
\]

\[
\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i + \sum_{i=1}^{k} b_{ii} x_i^2 + \sum_{i=1}^{k} \sum_{j=i+1}^{k} b_{ij} x_i x_j
\]

The parameters of the polynomials in Equations 4 and 5 are usually determined using a least squares regression analysis to fit these response surface approximations to existing data. These approximations are normally used for prediction within *Response Surface Methodology*. A more complete discussion of response surfaces and least squares fitting can be found in (Myers and Montgomery, 1995). An overview of RSM is given in Section 2.3.

#### 2.2.2 Neural Networks

A neural network is composed of neurons (single-unit perceptrons) which are multiple linear
regression models with a nonlinear (typically sigmoidal) transformation on y. If the inputs to each neuron are denoted \(x_1, x_2, \ldots, x_n\), and the regression coefficients are denoted by the weights, \(w_i\), then the output, \(y\), might be given by

\[ y = \frac{1}{1 + e^{-\sum w_i x_i + \beta}} \]

where \(\beta = \sum w_i x_i + \beta\). A neural network is then created by assembling the neurons into an architecture; the most prevalent is the multi-layer feedforward architecture.

There are two main issues in building a network: 1) specifying the architecture, and 2) training the network to perform well with reference to a training set. "To a statistician, this is equivalent to (i) specifying a regression model, and (ii) estimating the parameters of the model given a set of data." (Cheng and Titterington, 1994). If the architecture is made large enough, a neural network can be a nearly universal approximator (Rumelhart, et al., 1994).

"Training" a neural net refers to determining the proper values of all the weights (\(w_i\)) in the architecture, and is accomplished most commonly through backpropagation (Rumelhart, et al., 1994). First, a set of \(p\) training data points is assembled \(\{(x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p)\}\). For a network with a single output \(y\), the performance of the network is then defined as

\[ E = \sum_p (y_p - \hat{y}_p)^2 \]

where \(\hat{y}_p\) is the output that results from the network given input \(x_p\) and \(E\) is defined as the total error of the system. The weights are then adjusted in proportion to

\[ \frac{\partial E}{\partial y} \frac{\partial y}{\partial w_j} \]

Neural networks are best suited to approximate deterministic functions in regression-type applications. Cheng and Titterington (1994) note that "In most applications of neural networks that generate regression-like output, there is no explicit mention of randomness. Instead, the aim is function approximation." Typical applications are speech recognition and handwritten character recognition; very often the data is complex and of high dimensionality. Networks with tens of thousands of parameters are not unheard of, and the amount of training data is similar. Gathering the training data and determining the model parameters is a process that can be very computationally expensive.

\subsection{2.2.3 Rule Induction}

Rule induction is one of five main paradigms of machine learning that also include neural networks, case-based learning, genetic algorithms, and analytic learning (Langley and Simon, 1995). Of the five, inductive learning is most akin to regression and metamodeling and is therefore our focus here. An inductive learning system induces rules from examples, so the fundamental modeling constructs are condition-action rules. These rules in essence partition the data into discrete categories, and can be combined into decision trees for ease of interpretation. Many of the applications of inductive learning have been in process control and diagnostic systems, and inductive learning approaches can be used to automate the knowledge-acquisition process of building expert systems. Excellent examples can be found in (Evans and Fisher, 1994; Langley and Simon, 1995; Leech, 1986).

The training data collected are in the form of \(\{(x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p)\}\) where each \(x_1\) represents a vector of attribute values (such as processing parameters and environmental conditions), and each \(y_1\) represents a corresponding observed output value. Although attributes and outputs can be real-valued, the method is better suited to discrete-valued data. Real values must often be transformed into discrete representations (Evans and Fisher, 1994). Once the data set has been collected, training algorithms build a decision tree by selecting the "best" divisive attribute and then recursively calling the resulting data subsets. Although trees can be built by selecting attributes randomly, a more efficient approach is to select the attribute that minimizes the amount of information needed for category membership. The mathematics of such an information-theoretic approach are given in (Evans and Fisher, 1994).

Although decision trees seem best-suited for applications with discrete input and output values, there are also applications with continuous variables that have met with greater success than standard statistical analysis. For example, Leech (1986) reports a process-control application where "Standard statistical analysis methods were employed with limited success. Some of the data were non-numerical, the dependencies between variables were not well understood, and it was necessary to simultaneously control several characteristics of the final product while working within system constraints. The results of the statistical analysis, a set of correlations for each output of interest, were difficult for people responsible for the day-to-day operation to interpret and use."

\subsection{2.2.4 Kriging}

Since many computer analysis coedes are deterministic and are therefore not subject to measurement error, the usual measures of uncertainty derived from least-squares residuals have no obvious statistical meaning (Sacks, et al., 1989b). Consequently, some statisticians (e.g., (Sacks, et al., 1989a; Welch, et al., 1992)) have suggested modeling the response as a combination of a polynomial model (usually linear) plus departure, i.e.,

\[ y(x) = \sum_{j=1}^{k} \beta_j f_j(x) + Z(x) \]

where \(Z(x)\) is the systematic departure from the assumed model. \(Z(\bullet)\) represents the realization of a stochastic process and is assumed to have mean zero and covariance

\[ V(w, x) = \sigma^2 R(w, x) \]

between \(Z(w)\) and \(Z(x)\) where \(\sigma^2\) is the process variance and \(R(w, x)\) is the correlation. The covariance structure of \(Z\) relates to the smoothness of the approximating surface.

One method of analysis for such models is kriging (Matheron, 1963) which works as follows (Sacks, et al., 1989b). Given a design \(S = \{s_1, s_2, \ldots, s_n\}\) and data \(y_s = \{y(s_i)\}, \ldots, s_n\),
,..., \(y(s_i)\)'s, we are concerned with estimating the \(y(x)\) at an untried \(x\) using the linear (or polynomial) predictor

\[ \hat{y}(x) = c'(x) y_s. \]

If we consider \(\hat{y}(x)\) as random, \(y_s\) becomes \(Y_s = \{Y(s_i)\}, ..., Y(s_k)\}'s, and we can compute the mean squared error of the predictor averaged over the random process. Thus, the best linear unbiased predictor (BLUP) is found by picking the \(n \times 1\) vector \(c(x)\) to minimize

\[ \text{MSE}[\hat{y}(x)] = E[c'(x) Y_s - Y(x)]^2 \]

subject to the constraint for unbiasedness

\[ E[c'(x) Y_s] = E[Y(x)] \]  \(\text{(6)}\)

\(R(w,x)\) must first be specified in order to compute Equation 6 and should reflect the characteristics of the computer code. For a smooth response, a covariance function with some derivatives might be preferred, whereas an irregular response might call for a function with no derivatives. Sacks, et al. (1989b) discuss several possible correlation functions which result in a linear interpolation or cubic spline interpolation of the data (see Mitchell, et al., 1988) for more details). Consequently, kriging is extremely flexible due to the wide range of correlation functions \(R(w,x)\) which may be chosen. Depending on the choice of the correlation function, kriging can either "honor the data", providing an exact interpolation of the data, or "smooth the data", providing a nonexact interpolation of the data (Cressie, 1988). More information about kriging and the notion of modeling deterministic computer experiments as the realization of a stochastic process can be found in (Cressie, 1989; Journel, 1986; Laslett, 1994; Matheron, 1963; Sacks, et al., 1989b). Cressie (1988) deals specifically with predicting noiseless versions of \(Z\), as is the case for deterministic computer experiments. As a final note, keep in mind that kriging and splines (nonparametric regression models) are not the same. In fact, in several comparative studies, kriging has been seen to outperform splines and never performs worse than them (Laslett, 1994).

2.3 Strategies for Experimentation and Metamodeling

Two widely used methods incorporating experimental design, model building, and prediction are Response Surface Methodology and Taguchi's robust design or parameter design. A brief overview of these two approaches is provided.

2.3.1 Response Surface Methodology. Different authors describe Response Surface Methodology differently. Myers, et al. (1989) define RSM as "a collection of tools in design or data analysis that enhance the exploration of a region of design variables in one or more responses," Box and Draper (1987) state that, "Response surface methodology comprises a group of statistical techniques for empirical model building and model exploitation. By careful design and analysis of experiments, it seeks to relate a response, or output variable to the levels of a number of predictors, or input variables, that affect it." Finally, Biles (1984) defines RSM as the, "body of techniques by which one experimentally seeks an optimum set of system conditions".

RSM then encompasses and incorporates the design of experiments (Section 2.1), response surface model building (Section 2.2.1), and 'model exploitation' for exploring a factor space and seeking optimum factor settings. The general RSM approach includes all or a subset of the following steps:

i) screening: when the number of factors is too large for a comprehensive exploration and/or when experimentation is expensive, screening experiments are used to reduce the set of factors to those that are most important to the response(s) being investigated;

ii) first-order experimentation: when the starting point is far from an optimum (or in general when knowledge about the space being investigated is sought), first order-models and an approach such as steepest-ascent are employed to "rapidly and economically move to the vicinity of the optimum" (Montgomery and Evans, 1975);

iii) second-order experimentation: after the best solution using first-order methods is obtained, a second-order model is fit in the region of the first-order solution to evaluate curvature effects and attempt to improve the solution.

A more detailed description of RSM techniques and tools can be found in (Box and Draper, 1987), and a comprehensive review of RSM developments and applications from 1966-1988 is given in (Myers, et al., 1989).

2.3.2 Taguchi's Robust Design. Genichi Taguchi developed an approach for industrial product design that is built on the foundation of statistically designed experiments. Taguchi's robust design for quality engineering includes three steps: system design, parameter design, and tolerance design (Byrne and Taguchi, 1987). The key element of the approach is the parameter design step, within which statistical experimentation is incorporated.

Rather than simply improving or optimizing a response value, the focus in parameter design is to identify the factor settings that minimize variation in performance as well as adjust the mean performance to a desired target. Factors included in experimentation include control factors and noise factors. Control factors are those which can be set and held at specific values, while noise factors are those which cannot be controlled, e.g., temperature on a shop floor. The evaluation of mean performance and performance variation is accomplished through "crossing" two orthogonal arrays (Section 2.1.1). Control factors are varied according to an inner array, or "control" array, and for each run of the control array, the noise factors are varied according to an outer array, or "noise" array. For each control factor experiment, a response value is obtained for each noise factor design point, and the mean and variance of the response (measured across the noise design points) can be calculated. The performance characteristic used by Taguchi is a signal-to-noise (S/N) ratio defined in terms of the mean and variance of the response. Several alternate S/N ratios are available based on whether lower, higher, or nominal response
values are desired; specific $S/N$ ratio formulations can be found in (Ross, 1988).

The Taguchi approach does not explicitly include model building and optimization. Analysis of experimental results is used to identify factor effects, plan additional experiments, and to set factor values. Comprehensive presentation and application of the Taguchi approach is given in (Phadke, 1989; Ross, 1988). Issues and techniques presented by Taguchi have been used extensively in engineering design, and in many cases incorporated within the traditional RSM for efficient, effective, and robust design, e.g., (Myers and Montgomery, 1995). These applications and associated implications for engineering design are discussed in the next section.

3 METAMODELING IN THE CONTEXT OF ENGINEERING DESIGN

How are the metamodeling techniques of the previous section employed in engineering design? They can all be used to produce approximations of existing computer analysis codes, thereby producing fast analysis modules for more efficient computation. These metamodeling techniques can also yield insight to the functional relationship between input and output parameters.

![Figure 3. A Principal Use of Statistics in Computer-Oriented Design](image)

Where would such models be useful? There is an implicit process that is common to most of the engineering design applications, and this process is illustrated in Figure 3. A designer's ultimate goal is most often to arrive at improved or robust solutions, which are in essence the values of design variables that best meet the design objectives. (In statistical terms, the design variables are factors, and the design objectives are responses.) An intelligent search for these solutions most commonly relies on some choice of optimization techniques, which generate and evaluate many potential solutions in the path towards improvement. Regardless of the optimization technique employed a large number of function evaluations are required, thereby making fast analysis modules a near imperative.

When are metamodels useful or appropriate? In the later stages of design, when detailed information about a specific solution is available, highly accurate solutions are essential for measuring system performance. In the early stages of design, however, the focus is on generating, evaluating, and comparing potential conceptual configurations. In this stage, a conceptual design problem is characterized by a large amount of information, often uncertain, that must be managed. To ensure the identification of a "good" system level conceptual or preliminary configuration, a comprehensive search is necessary. In this case, the tradeoff between accuracy and efficiency becomes appropriate. The creation of metamodels allows fast analysis, which facilitates both comprehensive and efficient design space search, at the expense of a (hopefully slight) loss of accuracy.

Having established this baseline engineering design "context", in the next section we present a review of several statistical applications in engineering design, highlighting general trends and observations. In Section 3.2 we discuss a few of the more general statistical methods or "tools" which have been developed for engineering applications, and we conclude with Section 3.3 in which we take a closer look at the some of the statistical pitfalls associated with the application of statistical techniques to deterministic computer experiments. This then paves the way for Section 4, where we present our guidelines and recommendations for the appropriate use of statistics in computer-oriented design.

3.1 Existing Applications in Engineering Design

In this section we provide a survey of several engineering applications of Design of Experiments, Response Surface Methodology, and Taguchi's robust design method. Most of our examples come from aerospace and mechanical engineering design applications presented at conferences in recent years; for brevity's sake, we summarize our findings in Table 1. Our observations follow.

- In terms of experimental design, central composite designs and D-Optimal designs seem to be the preferred choice among aerospace engineers while orthogonal arrays (OAs) are preferred by many mechanical engineers; very few people utilize grid searches and random point generation since they are less efficient.

- Optimization seems to be the principal driver for aerospace applications of DOE and RSM; these types of design applications typically involve the use of computer intensive analysis and optimization routines, and RSM is a logical choice for increased efficiency.
• Mechanical engineers are more inclined to use OAs and Taguchi for robust design instead, working with the signal-to-noise ratio for parameter and tolerance design.
• Very few designers actually model Taguchi’s loss function directly (e.g., Beard and Sutherland, 1993); many prefer to model the response instead.
• Most applications employ second order response surface models; there are only a few instances were higher order models (e.g., Venter, et al., 1996) and mixed polynomial models (e.g., Roux, et al., 1996) are employed.
• When orthogonal arrays are used, special care must be taken to avoid confounding main effects with interactions, providing the interactions are not assumed to be insignificant based on prior observations or experience. Some authors take this into account while others do not.
• Most applications utilize a least squares regression analysis when fitting a model; only a few use stepwise regression, and this is usually because the model is not second order.

In addition to these specific engineering design applications of DOE, RSM, and Taguchi, many methods and “tools” have been developed for more general design application. We examine several of these methods in the next section, highlighting various applications of each.

3.2 Existing Methods and Tools in Engineering Design

In this section we present a few methods and tools developed specifically for engineering design that are build around or incorporate the statistical techniques and approaches reviewed in Section 2. As evident in Table 1 in the previous section, the Taguchi approach and RSM have been widely applied in engineering design; a review of literature comparing these approaches is given first, followed by an overview of Robust Design Simulation (RDS), Robust Concept Exploration Method (RCEM), NORMAN/DEBORA, DOE/Opt, and HIDER.

3.2.1 Taguchi Approach vs. RSM. Both Taguchi techniques and Response Surface Methodology have been applied extensively in engineering design. Many authors have compared these techniques for different problems. It appears to be commonly accepted that the principles associated with Taguchi approach are not only useful but very appropriate for industrial product design; Ramberg, et al. (1991) for example suggest that “the loss function and the associated robust design philosophy provide fresh insight into the process of optimizing or improving the simulation’s performance”. However, two specific aspects of the Taguchi approach are commonly criticized: the experimental design choice (orthogonal arrays, inner and outer) and the use of the loss function or signal-to-noise ratio. It has been argued and demonstrated that the use of a single experiment combining control and noise factors, a more traditional RSM approach rather than inner “control” and outer “noise” arrays, is more efficient in that it requires fewer experiments (Shoemaker, et al., 1991; Unal, et al., 1994a; Welch, et al., 1990). With respect to the loss function or signal-to-noise ratio, the drawbacks of combining response mean and variance into a single measure, a signal-to-noise ratio, are well-documented in the literature and many authors advocate measuring a response directly rather than loss, and separately tracking mean and variance (Chen, et al., 1995; Ramberg, et al., 1991; Welch, et al., 1990). Shoemaker et al. (1991), however, warn that “A potential drawback of the response-model approach that is depends more critically than the loss-model approach on how well the model fits.”

Given the general acceptance of the Taguchi principles and of robust design, with these criticisms of these techniques, many advocate a combined Taguchi-RSM approach, or simply applying more traditional RSM techniques within the framework defined by Taguchi (Lucas, 1994; Myers, et al., 1989; Myers and Montgomery, 1995; Ramberg, et al., 1991). It is the belief of the authors of this paper that both orthogonal, inner and outer arrays, and single, composite experiments each have advantages and disadvantages and appropriate uses, and that separate observation of mean and variance leads to useful problem insight. Regardless, the core principles of both Taguchi and RSM have provided a foundation for the specific design methods discussed in Section 3.2.2.

Table 1. Cited Engineering Applications of DOE, RSM, and Taguchi

<table>
<thead>
<tr>
<th>Paper</th>
<th>Experimental Design</th>
<th>Response Surface Method</th>
<th>Taguchi Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Balabany, et al., 1996)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Beard and Sutherland, 1993)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Chen, et al., 1996)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Chi and Bloebaum, 1995)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Engelund, et al., 1993)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Gadallah and ElMaraghy, 1994)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Giunta, et al., 1994)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Healy, et al., 1975)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Hong, et al., 1994)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Koch, et al., 1996)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Korngold and Gabriele, 1995)</td>
<td>x</td>
<td>2nd</td>
<td>x</td>
</tr>
<tr>
<td>(Li, et al., 1994)</td>
<td>x</td>
<td>2nd</td>
<td>x</td>
</tr>
<tr>
<td>(Mavis, et al., 1996)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Mavis, et al., 1995)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Roux, et al., 1996)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Venter, et al., 1996)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
<tr>
<td>(Yu and Iohn, 1993)</td>
<td>x</td>
<td>x</td>
<td>2nd</td>
</tr>
</tbody>
</table>

Copyright © 1997 by ASME
3.2.2 An Overview of Existing Methods. Robust Design Simulation (RDS), combines the design of experiments and response surface metamodeling aspects of RSM with Monte Carlo simulation to facilitate robust design (Mavris, et al., 1995). The general implementation of this approach consists of three sequential experimentation stages: screening, second-order experimentation for model building, and a combined experimentation/Monte Carlo simulation stage for incorporating noise. The objective of RDS is to achieve robust system designs: those that minimize performance variation due to uncontrollable noise, while achieving as closely as possible performance target values. RDS is presented and applied to the design of a High Speed Civil Transport aircraft in (Mavris, et al., 1996; Mavris, et al., 1995).

The Robust Concept Exploration Method (RCEM) has been developed to facilitate quick evaluation of different design alternatives and generation of top-level design specifications with quality considerations in the early stages of design (Chen, et al., 1996a). Foundational to the RCEM is the integration of robust design principles, design of experiments and response surface methodology, and the compromise Decision Support Problem (DSP) (a multiobjective decision model which is a hybrid formulation based on Mathematical Programming and Goal Programming). This RCEM is applied to a High Speed Civil Transport (HSTC) in (Chen, et al., 1996a). The authors of this paper have also applied RCEM to the design of a General Aviation Aircraft (Simpson, et al., 1996), to a manufacturing simulation problem (Peplinski, et al., 1996), and to the design of a turbine lift engine (Koch, et al., 1996).

NORMAN/DEBORA is a TCAD (Technology Computer Aided Design) system created by a Belgian company, IMEC, incorporating advanced sequential design of experiments (DOE) and Response Surface Methodology (RSM) techniques to aid in engineering optimization and robust design problems. Advanced DOE/RSM techniques incorporated within this system include a novel design of experiments concept, Target Oriented Design (TOD), a unique parameter transformation technique, RATIOFIND, and a non-linearly constrained optimizer, DEBORA (Booth, et al., 1991). The NORMAN/DEBORA system is demonstrated for semiconductor integrated circuit (IC) technology design and optimization in (Cartuyvels and Dupas, 1993).

DOE/Opt is a prototype computer system for performing design of experiments, response surface modeling, and optimization (Boning and Mozumder, 1994). Applications described for DOE/Opt are in the semiconductor process and device design area and include process/device optimization, simulator tuning, process control recipe generation, and design for manufacturability. Hierarchical and Interactive Decision Refinement (HIDER) is a methodology for concept exploration in the early stages of design that integrates simulation, optimization and machine learning to support design decision making (Reddy, 1996). HIDER uses the Adaptive Interactive Modeling System (AIMS) (Lu and Tcheng, 1991) to decompose the design space using distance-based, population-based, and hyperplane-based algorithms. HIDER and AIMS have been applied to the design of a cutting process (Lu and Tcheng, 1991), a diesel engine (Yerramreddy, et al., 1992), and a wheel loader (Reddy, 1996).

The Concurrent SubSpace Optimization (CSSO) methodology developed in (Renaud and Gabriele, 1993; Renaud and Gabriele, 1994; Renaud and Gabrielle, 1991) utilizes design data generated during concurrent subspace optimizations to develop response surface approximations of the design space. Optimization of these response surface representations of the design space form the basis of the coordination procedure developed in (Renaud and Gabriele, 1993; Renaud and Gabriele, 1994; Renaud and Gabrielle, 1991). The data generated by the subspace optimizers is not uniformly centered about the current design as in CCD or other sampling strategies, but instead follows the descent path of the subspace optimizers. In (Renaud and Gabriele, 1993; Renaud and Gabriele, 1994; Renaud and Gabrielle, 1991) interpolating polynomial response surfaces are constructed which have either a first order or second order basis for use in the CSSO coordination procedure. In (Wujek, et al., 1996; Wujek and Renaud, 1997) a modified decomposition strategy is employed in developing quadratic response surfaces for use in the CSSO coordination procedure. Finally, in (Sellar and Batill, 1996; Sellar, et al., 1996a; Sellar, et al., 1996b) artificial neural network response surfaces are developed for use in the CSSO coordination procedure.

Additional approaches incorporating statistical techniques in design exist; only a few have been included in this section. Our focus is not on the methods, but on the appropriateness of the statistical techniques: many of the examples to which these methods have been applied employ deterministic computer experiments, to which the application of statistical techniques is questionable. The associated issues are discussed in the next section.

3.3 A Closer Look at Experimental Design for Deterministic Computer Experiments

Since engineering design commonly involves exercising deterministic computer codes, the use of statistical experimentation in creating metamodels of these codes warrants a closer look. Given a response of interest, y, and a vector of independent factors x thought to influence y, the relationship between y and x (Equation 3) includes the random error term ε. To apply least squares regression, the error values for each data point are assumed to have identical and independent normal distributions with means of zero and standard deviations of σ, or ε, i.i.d. N(0,σ²). This scenario is shown in Figure 4(a). The least squares estimator then minimizes the sum of the squared differences between the actual data points and the values predicted by the model. It is acceptable if no data point actually lies on the predicted model, because it is assumed that the model "smooths out" the random error. Of course, it is likely that the regression model itself is only an approximation of the true behavior between x and y, so that the final relationship is

\[ y = g(x) + \varepsilon_{\text{bias}} + \varepsilon_{\text{random}} \] (7)
4(b), $\epsilon_{\text{random}}$ has mean zero and variance zero, so after model fitting we have the relationship

$$y = g(x) + \epsilon_{\text{bias}}$$

(8)

The deterministic case of Equation 7 conflicts sharply with the methods of least squares regression. First, unless $\epsilon_{\text{bias}}$ is i.i.d. $N(0, \sigma^2)$ then the assumptions for statistical inference from least squares regression are violated. Even further, because there is no random error there is little justification for smoothing across data points; instead the model should hit each point exactly and interpolate between them as shown in Figure 4(b). Finally, most standard tests for model and parameter significance are based on computations using $\epsilon_{\text{random}}$ (the mean squared error) and are therefore impossible to compute. These observations are supported by literature in the statistics community; as Sacks, et al. (1989b) carefully point out, because deterministic computer experiments lack random error:

- response surface model adequacy is determined solely by systematic bias,
- the usual measures of uncertainty derived from least-squares residuals have no obvious statistical meaning (deterministic measures of uncertainty exist, e.g., $\text{max} | y(x) - y(x) \text{ over } x$, but they may be very difficult to compute), and
- the classical notions of experimental blocking, replication and randomization are irrelevant.

Similarly, according to Welch and his co-authors (1990), current methods for the design and analysis of physical experiments (e.g., Box and Draper, 1987; Box, et al., 1978) are not ideal for complex, deterministic computer models. "In the presence of systematic error rather than random error, statistical testing is inappropriate" (Welch, et al., 1990). Finally, a discussion of how the model should interpolate the observations can be found in (Sacks, et al., 1989a).

4 GUIDELINES AND RECOMMENDATIONS

How can a design engineer efficiently apply the metamodeling tools of Section 2 while avoiding the pitfalls described in Section 3.3? There are two ways to answer this question: from the bottom up (tools applications) and from the top down (motives tools). The bottom-up approach is presented in Section 4.1, the top-down in Section 4.2.

4.1 Evaluation of Metamodeling Techniques

There are two components to this section. The first is an evaluation of the four metamodeling techniques described in Section 2.2. The second component is choosing an experimental design which has more direct applicability to response surface methods but also applies to the remaining metamodeling techniques. The explorations of which designs are most appropriate are, as yet, open research areas.
4.1.1 Evaluation of Model Choice and Model Fitting Alternatives. In Section 2.2 we described four metamodeling techniques; here we present some brief guidelines for their evaluation.

Response Surfaces: primarily intended for applications with random error, though deterministic applications are not necessarily incorrect. Building higher-order models with greater than ten factors becomes difficult. (Building regression models is tractable for fifteen to twenty factors, but actually obtaining the data is the limiting factor.) It is the most well-established metamodeling technique, and is probably the easiest to use.

Neural Networks: nonlinear regression approach best suited for deterministic (non-noisy) applications. They can be a nearly universal approximator, and so are able to handle highly nonlinear or extremely large (~ 10,000 parameters) problems, as long as enough data points are provided and enough computer time is allocated to fit the model. Neural networks therefore lend themselves to large problems, so that "Now the procedure is to toss the data directly into the NN software, use tens of thousands of parameters in the fit, let the workstation run 2-3 weeks grinding away doing the gradient descent, and voilà, out comes the result" (Cheng and Titterington, 1994).

Inductive Learning: modeling technique most appropriate where the input and output factors are primarily discrete-valued or can be grouped into categories. The predictive model, in the form of condition/action rules or a decision tree, may perhaps lack the mathematical insight desired for engineering design applications, but it is ideal for applications such as process control and diagnosis.

Kriging: interpolation method created to handle deterministic data. It is not based on the same assumptions as standard regression analysis and appears to be better suited for deterministic computer experiments, as long as the number of factors is small (≤10, according to (Welch, et al., 1992)). Kriging is extremely flexible due to the wide scope of correlation functions R(w,x) which may be chosen (Laslett, 1994); it can provide an exact interpolation of the data or "smooth the data" (Cressie, 1988). Additional advantages include (Welch, et al., 1992):

• it provides a basis, via the likelihood, for a stepwise algorithm to determine the important factors;
• it is flexible, allowing nonlinear and interactions effects to emerge without explicitly modeling such effects;
• the same data can be used for screening and building the predictor, so expensive runs are efficiently used.

The complexity of the method, however, coupled with a lack of computer software support, may make the learning curve of this technique prohibitive in the near term.

4.1.2 Evaluation of Experimental Designs. There are many voices in the discussion of the relative merits of different experimental designs, and it is therefore unlikely that we have captured them all. However, broad trends appear, and in general we have found that central composite designs consistently perform and compare favorably. Other second order designs worth investigating are the hexagonal (Montgomery and Evans, 1975) and Box-Behnken and hybrid designs (Giovanetti-Jensen and Myers, 1989). For comparisons of different design types, refer to the following.

- Lucas (1976) compares CCD, Box-Behnken, uniform shell, Hoke, Pesotchinsky, and Box-Draper designs, using the D-efficiency and G-efficiency statistics.
- Myers and Montgomery (1995) provide a comprehensive review of many experimental designs for fitting second order response surfaces. They conclude that hybrid designs are a good idea, if the unusual levels for the design variables can be tolerated. For computer experiments, this is unlikely to be a problem.
- Montgomery and Evans (1975) compare six second-order designs: a) 3^2 factorial, b) rotatable orthogonal CCD, c) rotatable uniform precision CCD, d) rotatable minimum bias CCD, e) rotatable orthogonal hexagon, and f) rotatable uniform precision hexagon. The criteria used for comparison are average response achievement and distance from true optimum.
- Giovanetti-Jensen and Myers (1989) discuss several first and second order designs, observing that the performance of rotatable CCD and Box-Behnken designs are nearly identical. They note that "hybrid designs appear to be very promising."

4.2 Initial Recommendations for Metamodeling Uses

The majority of metamodeling applications are built around the creation of low-order polynomial metamodels using central composite designs and least squares regression. The nearly universal popularity of this approach is due, at least in part, to the maturity and well-established nature of response surface methodology, the ease and simplicity of the method, and easily accessible software support tools. However, the RSM approach starts to break down when there are a large (> 10) number of factors to be modeled, or when the relationship to be modeled is highly nonlinear. And as we have shown in Section 3.3, there are also dangers of applying the RSM approach blindly to deterministic applications. The alternative approaches to metamodeling described in section 4.1.1 address these limitations, each in their own way. Our recommendations are summarized as follows:

- If a large number of factors must be modeled in a deterministic application, then neural networks may be the best metamodeling choice despite their tendency to be computationally expensive to create.
- If the underlying function to be modeled is deterministic and highly nonlinear in a small (≤10) number of factors, then kriging may be the best choice despite its statistical complexity.
- However, in deterministic applications that have a small number of factors and are fairly well behaved, we present another option for exploration: applying the standard RSM approach, augmented with a Taguchi outer array.
**RSM/OA approach:** The fundamental problem with applying least-squares regression to deterministic applications is the absence of \( \epsilon_{\text{random}} \) in Equation 8. However, if some of the input parameters for the computer code can be classified as noise factors, and if these noise factors are varied across an outer array for each setting of the control factors, then in essence a series of replications are generated that can be used to approximate \( \epsilon_{\text{random}} \). This approximation can be justified if it is reasonable to assume that, were the experiments performed on an actual physical system, that the random error observed would have been due to fluctuations in the same noise factors that are varied in the computer code. If this assumption is reasonable, then statistical testing of model and parameter significance can be performed, and models of both mean response and variability can be created from the same experimental design. We are currently investigating this approach in our ongoing work.

5 SUMMARY AND CONCLUSIONS

In this paper we have surveyed some applications of statistics to engineering design and have developed the idea of metamodelling in the engineering design context, as discussed in Section 1 and as shown in Figure 3. Metamodelling is an active research area in more than just the statistics community, to which our reference list is mute testimonial. However, applying these techniques to deterministic applications in engineering design can lead to problems, as pointed out in Section 3.1 and discussed in Section 3.3. We present the set of our recommendations for applying metamodelling techniques in Section 4, but this set is by no means complete. Comprehensive comparisons of these techniques have yet to be performed, and the difficulties of large problem size and nonlinearity are ever-present.

In particular, an issue of interest to us in our continuing research is the problem of size, or scaling. As the size of a problem (number of factors) increases, the cost associated with creating metamodels can begin to outweigh the associated gains. In addition, often the reduction in factors resulting from screening is still not sufficient to bring the problem to a manageable size. This problem is compounded by the multiple response problem: complex engineering design problems invariably include multiple measure of performance, or responses, that must be modeled. The screening process also breaks down when attempting to select the most important factors for more than one response, as each response often requires different important factors. The general question that arises from these problems, then, is how can these experimentation and metamodelling techniques be used efficiently for larger problems (problems with greater than 10 factors after screening)? One approach that we are pursuing is problem partitioning or decomposition. A significant literature base exist in techniques for breaking a problem into smaller problems; a good review of such methods can be found in (Lewis and Mistree, 1997). Using these techniques, a complex problem may be broken down into smaller problems that allow efficient experimentation and metamodelling, which again will lead to comprehensive and efficient exploration of a design space. We hope to have preliminary results in this area in the near future.

**ACKNOWLEDGMENTS**

Timothy Simpson and Jesse Peplinski have been supported by NSF Graduate Research Fellowships, and Patrick Koch has been supported by a Gwaltney manufacturing fellowship. In addition, financial support from NSF Grants DMI-96-12365 and DMI-96-12327, and NASA Grant NGT-51102 is gratefully acknowledged.

**REFERENCES**


