

Metamodel-Based Parameter Design of Static Systems

Turuna S. Seecharan^{aΨ} and Gordon J. Savage^b

University of Waterloo, Department of Systems Design Engineering, Waterloo, Ontario, ON, Canada N2L 3G1

^aE-mail: tseechar@uwaterloo.ca

^bE-mail: gjsavage@uwaterloo.ca

^Ψ Corresponding Author

(Received 30 June 2011; Revised 24 October 2011; Accepted 21 November 2011)

Abstract: When attempting to achieve an improved design of some physical system, the important variables, the “design variables”, are manipulated to find the combination that results in the best system performance. Most research deals with cases where design variables are deterministic thus ignoring possible uncertainties present due to manufacturing or environmental conditions. When uncertainty is considered, the design variables follow a particular distribution whose parameters are defined. Parameter design aims to reduce the probability of failure of a system by moving the distribution parameters of the design variables. In this paper, we focus on moving the means of the design variables to search for a reduced failure probability. The most popular method to estimate the probability of failure is a Monte Carlo Simulation where, using the distribution parameters, many design variable combinations are generated and the number of times the corresponding response does not meet specifications is counted. This method, however, can become time-consuming as mechanistic models become increasingly complex. From structural reliability theory, the First Order Reliability Method (FORM) is an efficient method to estimate probability and to search for the parameters that reduce the failure probability. However, if the mechanistic model is too complex or implicit, FORM becomes difficult to use. This paper presents a methodology that uses approximating functions, called ‘metamodels’, with FORM to search for the design parameters that minimises the probability of failure. The method will be applied to three examples and the accuracy, along with the speed, of the methodology will be discussed.

Keywords: Metamodel, Response Surface Model, Kriging, First-Order Reliability Method

Notation

$f(\mathbf{v})$	A function of the design variables, \mathbf{v} .	β_{R_k}	Reliability Index corresponding to the k^{th} LSF.
$g_k(\mathbf{u})$	k^{th} LSF (in \mathbf{u} -space).	$\beta_0, \beta_1, \dots, \beta_l$	Metamodel parameters.
$g_k(\mathbf{v})$	k^{th} LSF (in \mathbf{v} -space).	$\frac{\partial y(\mathbf{v})}{\partial v_i}$	First derivative of the metamodel with respect to the i^{th} design variable.
n	Total number of design variables.	$\theta_1, \dots, \theta_p$	Kriging correlation function parameters.
N_{LF}	Total number of LSFs.	ζ	Upper or Lower Design Specification.
$y(\mathbf{v})$	Metamodel	Φ	Normal CDF Function.
M	Total number of training sets.	Γ	Kriging model correlation matrix whose elements represent the correlation between training sets.
\mathbf{p}	Vector of design parameters, $\mathbf{p} = [\mu_1, \dots, \mu_n]^T$.	CV	Cross-Validation.
$\mathbf{r}(\mathbf{v})$	Column vector of functions of the design variable.	CV-RMSE	Cross-Validation Root Mean Squared Error.
$\tilde{\mathbf{r}}^T(\mathbf{v})$	Vector of the correlation between the new response, \mathbf{v} , and each training set.	FORM	First Order Reliability Method.
\mathbf{u}_k^*	MLFP corresponding to the k^{th} LSS.	LSF	Limit State Function.
\mathbf{v}	Vector of design variables.	LSS	Limit State Surface.
\mathbf{w}^T	Vector of metamodel constants.	MCS	Monte Carlo Simulation.
\mathbf{x}_i	The vector of training sets for the i^{th} design variable.	MLFP	Most Likely Failure Point.
\mathbf{z}	Vector of training responses.	RSM	Response Surface Model
\mathbf{X}	Matrix of training design variable sets.		
α_k	$\text{sign}(g_k(\mathbf{u}) = 0)$.		
β_R	Reliability Index.		

1. Introduction

For most physical systems, experimentation is costly especially in the case of design where the experimental space is searched to find the best design (Barton, 1998). To compensate, analysts usually attempt to find a mathematical model to describe the behaviour of the system (Esfandiari and Lu, 2010; Lobontiu 2010). Such

models are termed, herein, as “mechanistic models”. Using these models, computer simulations are built to simulate the behaviour of the physical system. In the design of systems, much research involves deterministic optimisation that does not account for uncertainties present in manufacturing or environment conditions. More recent research, however, attempts to account for

such uncertainties by assuming that the design variables follow some type of distribution (Grandhi and Wang, 1998; Allen et al., 2004).

Reliability-Based Design Optimisation deals with problems with uncertainty present in design variables. Here, the problem usually entails minimising a cost while ensuring certain probabilistic constraints are met. These constraints relate the demand with the performance characteristic in the form of a limit state function (LSF) and the probability of failure is calculated using either the Monte Carlo Simulation (MCS) or the structural reliability concept the First Order Reliability Method (FORM) (Madsen et al., 1986).

In probabilistic design, when uncertainty is considered, we attempt to find the distribution parameters (means and/or tolerances) of the design variables that improve or optimise the selected performance measures. Furthermore, an optimum system performance occurs when the probability of failure is at a minimum. The design variables include for example sizes, weights and material compositions.

In most cases, however, the mechanistic model is too complex or implicit thus resulting in lengthy Monte Carlo simulations or difficult reliability calculations. These mathematical models are usually in the form of differential or even implicit equations thus making them difficult to use in simple design calculations. Recent research in function approximation and response estimation has developed approximating functions, known as “metamodel”, to replace complex mechanistic models (Barton, 1998; Hussain et al., 2002).

A lot of research has compared the performance of various metamodels in fitting complex or even implicit functions (Hussain et al., 2002; Jin et al., 2001; Simpson et al., 2001; Muller and Messac, 2006). These papers have discussed the advantages and disadvantages of different metamodels and their ability to fit functions of varying degrees of complexity.

Recent research has started using metamodels in reliability analysis (Schueremans and Van Gemert, 2005; Deng, 2006). In these problems, design variables are random and, provided with design criteria, the probability of failure is estimated using either Monte Carlo Simulation (Schueremans and Van Gemert, 2005) or the first- or second-order reliability method (FORM or SORM) (Deng, 2006). These papers have shown that metamodels offer accurate and efficient alternatives where mechanistic models are implicit or just too complex. This paper will extend this idea to use metamodels to search for the best design using FORM.

2. Proposed Methodology

This paper presents a methodology that uses metamodels and FORM to search for the means of the design variables that result in an improved system performance as applied to static systems. When we refer to a “static

system” we refer to a system with a steady-state response. This improved performance is based upon the specific design specifications that the system must meet. The flowchart shown in Figure 1 outlines the major steps involved in this metamodel-based parameter design methodology.

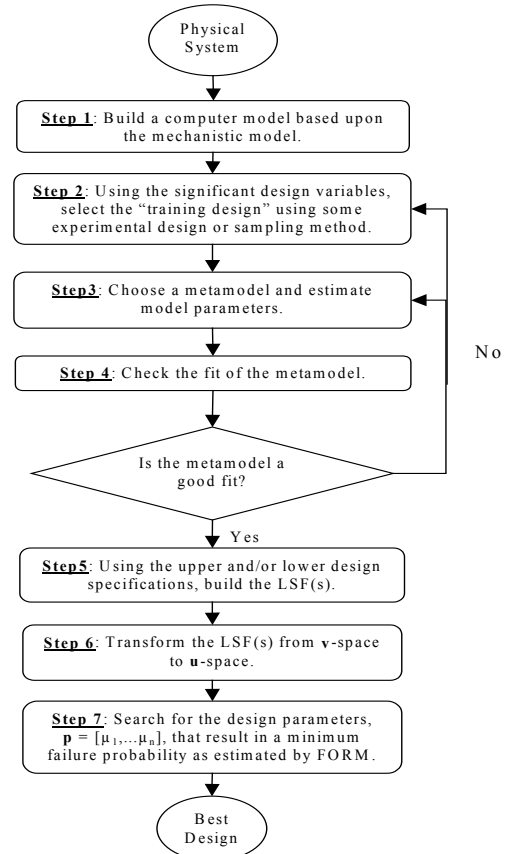


Figure 1. A flowchart showing the main steps involved in metamodel-based parameter design

Typically, we start with a mathematical representation of some physical system which we then use to build a computer-based simulation using a program like MATLAB[®]. The next step involves extracting a sample of experimental design variable sets and their corresponding responses, called the ‘training design’, which is used to estimate the parameters of the chosen metamodel. An important step in metamodel-based design is to check the fit of the metamodel. If the fit is not found to be acceptable, there are two choices we can make; the first is to use another metamodel or the second is to use another sampling method. If, however, the metamodel is a good fit, we then use the metamodel and the stated design criteria to build a “limit-state function”. This limit-state function, LSF, is used with the first-order reliability method, a method popular in reliability analysis, to estimate the failure probability when provided with the initial design parameters. These

design parameters are made up of the mean and tolerance of each design variable. For the purpose of this work, we will focus on the means. Using the LSF, FORM and an optimisation routine like “fmincon” in MATLAB®, the design parameters are moved so that the failure probability is reduced.

To illustrate the proposed methodology, three examples will be presented whose responses must meet specific design criteria. In the following sections, this paper will present an overview of two metamodels; Response Surface and Kriging Models. A brief overview will also be provided about FORM and will show how the metamodels can be used in FORM to calculate the probability of failure of some arbitrary system when provided with pre-stated design specifications. The method will be applied to three simple case studies and the speed and accuracy of this metamodel-based parameter design method will be discussed.

3. Metamodelling Process

In the modelling process, the modeller abstracts properties from various components in order to obtain a representation of the physical world. In a similar way, a model of the modelling process itself can be obtained; this is termed a metamodel (Barton, 1998). In the metamodelling process, three main steps are involved; first, a sample of design variables and their corresponding responses is obtained from the computer model, second, a metamodel form is chosen and using the “training design” obtained in Step 1, the metamodel parameters are estimated. Finally, the fit of the metamodel is checked.

3.1 The Training Design

The training design is a sample of responses from the computer simulation of the mechanistic model that is used to ‘fit’ the metamodel. It is chosen through the use of statistical design of experiments such as the Full or Fractional Factorial Designs or sampling techniques like Latin Hypercube Sampling (Montgomery, 2005). The design gives rise to a total of M runs that are used to generate some initial dynamic responses. The design variables used to generate these initial responses are called the “training sets”.

Using some experimental design technique or sampling method, M different combinations of n design variables are obtained and recorded into the matrix \mathbf{X}_d as shown. Each unique combination of the n design variables is termed the “training set”.

$$\mathbf{X}_d = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_M^T \end{bmatrix} = \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ x_{2,1} & \dots & x_{2,n} \\ \vdots & \ddots & \vdots \\ x_{M,1} & \dots & x_{M,n} \end{bmatrix}_{M \times n}$$

In \mathbf{X}_d , $x_{j,i}$ corresponds to the j^{th} training set of the i^{th} design variable ($j = 1, \dots, M$ and $i = 1, \dots, n$). The matrix

\mathbf{X}_d is then input to the computer simulation based on the mechanistic model and a sample of training responses, corresponding to each training set, is obtained.

$$\mathbf{z}^T = \left[z(\mathbf{x}_1^T) \quad z(\mathbf{x}_2^T) \quad \dots \quad z(\mathbf{x}_M^T) \right]_{1 \times M}$$

3.2 Metamodel Form

In general, the metamodels can have the form,

$$y(\mathbf{v}) = \mathbf{w}^T \mathbf{r}(\mathbf{v}) \quad \dots \text{Eq.1}$$

where $y(\mathbf{v})$ is a function of the response with respect to the design variables, \mathbf{v} . \mathbf{w} is a row vector of the metamodel constants estimated using the training data (\mathbf{X}_d, \mathbf{z}) and $\mathbf{r}(\mathbf{v})$ is a column vector whose elements contain functions of the design variable. At this point, clear distinction should be made between $x_{j,i}$, which is a sample point of variable i and v_i which is the general term. This is necessary since the gradient-based optimisation with FORM requires derivatives of $y(\mathbf{v})$ with respect to each variable v_i . Therefore, since the fitting parameters are constants, the derivatives of interest is

$$\frac{\partial y(\mathbf{v})}{\partial v_i} = \mathbf{w}^T \frac{\partial \mathbf{r}(\mathbf{v})}{\partial v_i} \quad \dots \text{Eq.2}$$

3.3 Response Surface Model (RSM)

Perhaps the simplest and most common metamodel is the Response Surface Model (RSM). Given a response, y , and a vector of design variables, \mathbf{v} influencing y , the relationship between y and \mathbf{v} is

$$y(\mathbf{v}, \boldsymbol{\beta}) = \beta_0 + \sum_{i=1}^n \beta_i v_i + \sum_{i=1}^n \beta_{ii} v_i^2 + \sum_{i < j} \beta_{ij} v_i v_j + \varepsilon \quad \dots \text{Eq.3}$$

where β_0, β_i, \dots are model parameters to be estimated and v_i represents the i^{th} design variable. In this paper, we use a second-order RSM. Equation 3 can be transformed to look like Equation 1 so that \mathbf{w} is a vector of constants and $\mathbf{r}(\mathbf{v})$ is a vector of functions of the design variables. An example of how this is done is shown in the Appendices A, B and C.

For a second-order RSM, the length of $\mathbf{r}(\mathbf{v})$ depends upon the total number of design variable, n through the equation

$$l = \prod_{j=0}^{d-1} \frac{n + (2 - j)}{2 - j} \quad \dots \text{Eq.4}$$

Therefore, when $n=2, l = 6$. The actual data matrix must have l columns and so \mathbf{X}_d is augmented by a unit vector to the left and an appropriate sub-matrix of functions of \mathbf{X}_d to the right, so

$$\mathbf{X}_a = \left[\mathbf{1} \quad \mathbf{X}_d \quad f(\mathbf{X}_d) \right]_{(M \times l)} \quad \dots \text{Eq.5}$$

where $f(\mathbf{X}_d)$ is a high order function (second-order for the purposes of this paper) of the training sets that follows the factorial arrangement. Therefore, for $n = 2$,

$$f(\mathbf{X}_d) = \begin{bmatrix} \mathbf{x}_1^2 & \mathbf{x}_2^2 & \mathbf{x}_1 \mathbf{x}_2 \end{bmatrix} \quad \text{and}$$

$$\mathbf{X}_a = [1 \quad \mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_1^2 \quad \mathbf{x}_2^2 \quad \mathbf{x}_1\mathbf{x}_2] .$$

Typically, for the RSM, model parameters are estimated using the least-squares estimation method

$$\hat{\boldsymbol{\beta}} = [\mathbf{X}_a^T \mathbf{X}_a]^{-1} \mathbf{X}_a^T \mathbf{z} \quad \dots \text{Eq.6}$$

To estimate the variance, σ^2 , consider the sum of squares of the residuals

$$SS_E = \sum_{i=1}^M (z_i - \hat{z}_i)^2 = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} \quad \dots \text{Eq.7}$$

Substituting $\boldsymbol{\varepsilon} = \mathbf{z} - \hat{\boldsymbol{\beta}}^T \mathbf{X}^T$ into Equation 7 and simplifying, the sum of squares of the residuals becomes

$$SS_E = \mathbf{z}^T \mathbf{z} - \hat{\boldsymbol{\beta}}^T \mathbf{X}^T \mathbf{z} \quad \dots \text{Eq.8}$$

with $M-(n+1)$ degrees of freedom. It can be shown that $E(SS_E) = \sigma^2(M-1)$ (Montgomery, 2005). Therefore, an unbiased estimator of σ^2 is given by

$$\hat{\sigma}_{RSM}^2 = \frac{SS_E}{M-1} \quad \dots \text{Eq.9}$$

When presented with a full second-order polynomial RSM, the significance of each model parameter can be tested to reduce the size of the model (Montgomery, 2005). The details of this method will not be explored in this paper but such a method is worth mentioning. Also differentiating the second-order RSM is not difficult. Therefore, the details are not shown.

3.4 Kriging

There are several types of Kriging models; ordinary, universal, detrended etc. In this work, we use a universal Kriging model in which the first part of the model is represented by a linear RSM. Given sampled inputs, \mathbf{X}_d , and the resulting outputs $\mathbf{z} = [z(\mathbf{x}_1), \dots, z(\mathbf{x}_M)]^T$, Kriging treats the linear predictor at any point \mathbf{v} , $y(\mathbf{v})$, as a random function (Sacks et al., 1989; Martin and Simpson, 2005). The metamodel is then the Best Linear Unbiased Predictor (BLUP) of $y(\mathbf{v})$

$$y(\mathbf{v}) = \beta_0 + \tilde{\boldsymbol{\beta}}^T f(\mathbf{v}) + \tilde{\mathbf{r}}^T(\mathbf{v}) \boldsymbol{\Gamma}^{-1}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}) \quad \dots \text{Eq.10}$$

where $\tilde{\boldsymbol{\beta}}^T = [\beta_1 \quad \dots \quad \beta_n]$, $f(\mathbf{v}) = [v_1 \quad \dots \quad v_n]^T$, $\boldsymbol{\beta} = [\beta_0 \quad \tilde{\boldsymbol{\beta}}^T]^T$ and $\mathbf{X} = [1 \quad \mathbf{X}_d]^T$. $\boldsymbol{\Gamma}$ is a special correlation matrix that quantifies the correlation between two observations, \mathbf{x}_1 and \mathbf{x}_2 , using a spatial correlation function $\gamma(\mathbf{x}_1, \mathbf{x}_2)$

$$\gamma(\mathbf{x}_j, \mathbf{x}_k) = \exp\left(-\sum_{i=1}^n (\theta_i (x_{j,i} - x_{k,i})^2)\right) \quad j, k = 1 \dots M \quad \dots \text{Eq.11}$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma(\mathbf{x}_1, \mathbf{x}_1) & \dots & \gamma(\mathbf{x}_1, \mathbf{x}_M) \\ \vdots & \ddots & \vdots \\ \gamma(\mathbf{x}_M, \mathbf{x}_1) & \dots & \gamma(\mathbf{x}_M, \mathbf{x}_M) \end{bmatrix}_{M \times M} \quad \dots \text{Eq.12}$$

where θ_i is the model parameter to be estimated. The vector, $\tilde{\mathbf{r}}^T$, contains the correlation between an unknown point, \mathbf{v} , and the M known sample points

$$\tilde{\mathbf{r}}^T(\mathbf{v}) = [\gamma(\mathbf{v}, \mathbf{x}_1), \dots, \gamma(\mathbf{v}, \mathbf{x}_M)]_{1 \times M} \quad \dots \text{Eq.13}$$

Each element in $\tilde{\mathbf{r}}^T, \tilde{r}_j$, is found using Equation 14.

$$\tilde{r}_j(\mathbf{v}) = \exp\left(-\sum_{i=1}^n \theta_i (v_i - x_{j,i})^2\right) \quad \dots \text{Eq.14}$$

Equation 10 can be simplified to the general metamodel of equation (1) where

$$\mathbf{w}^T = \left[\boldsymbol{\beta} \quad \left(\boldsymbol{\Gamma}^{-1}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}) \right)^T \right]_{1 \times (M+n+1)} \quad \text{and} \\ \mathbf{r}(\mathbf{v}) = \left[1 \quad f^T(\mathbf{v}) \quad \tilde{\mathbf{r}}^T(\mathbf{v}) \right]_{(M+n+1) \times 1}^T \quad \dots \text{Eq.15}$$

The Maximum Likelihood Estimation (MLE) method is most commonly used to provide the best estimate of the Kriging model parameters where the optimal value of $\boldsymbol{\beta}$ is found to be

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \boldsymbol{\Gamma}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \boldsymbol{\Gamma}^{-1} \mathbf{z} \quad \dots \text{Eq.16}$$

and $\hat{\sigma}_{KG}^2$ is

$$\hat{\sigma}_{KG}^2 = \frac{1}{M} (\mathbf{z} - \mathbf{X}\hat{\boldsymbol{\beta}})^T \boldsymbol{\Gamma}^{-1} (\mathbf{z} - \mathbf{X}\hat{\boldsymbol{\beta}}) \quad \dots \text{Eq.17}$$

Since a closed-form solution does not exist for $\boldsymbol{\theta}$ the best estimate for $\boldsymbol{\theta}$ is found by maximising Equation 18 (Martin and Simpson, 2005; Simpson et al., 2001).

$$L(\hat{\boldsymbol{\theta}}) = -\frac{[M \ln(\hat{\sigma}_{KG}^2) + \ln|\boldsymbol{\Gamma}|]}{2} \quad \dots \text{Eq.18}$$

3.5 Kriging Derivatives

After converting the general Kriging model to look like Equation 2, the model is differentiated, as in Equation 3, where

$$\frac{\partial \mathbf{r}(\mathbf{v})}{\partial v_i} = \left[0 \quad \frac{\partial f(\mathbf{v})}{\partial v_i} \quad \frac{\partial \tilde{\mathbf{r}}(\mathbf{v})}{\partial v_i} \right]^T \quad \dots \text{Eq.19}$$

Since $f(\mathbf{v})$ is just a linear function, differentiating this function is straightforward. However, the same cannot be said for $\tilde{\mathbf{r}}(\mathbf{v})$. Suppose we simplify equation 14, so that

$$\tilde{r}_j(\mathbf{v}) = \exp(-\rho_j(\mathbf{v})) \quad \dots \text{Eq.20}$$

Where $\rho_j(\mathbf{v}) = \sum_{i=1}^n \theta_i (v_i - x_{i,j})^2$. Then each element in

$\tilde{\mathbf{r}}(\mathbf{v})$ is differentiated as follows

$$\frac{\partial \tilde{r}_j(\mathbf{v})}{\partial v_i} = \frac{\partial \tilde{r}_j(\mathbf{v})}{\partial \rho_j(\mathbf{v})} \times \frac{\partial \rho_j(\mathbf{v})}{\partial v_i} \quad \dots \text{Eq.21}$$

After some simplification (see Appendix C), Equation 21 becomes

$$\frac{\partial \tilde{r}_j(\mathbf{v})}{\partial v_i} = -2\theta_i (v_i - x_{i,j}) r_j(\mathbf{v}) \quad \dots \text{Eq.22}$$

3.6 Metamodel Validation

After estimating metamodel parameters, the next step is

to determine how the metamodel fits the training design. There are two methods involved in checking the fit of a metamodel; the first involves estimating the ability of the metamodel to reproduce the responses of the training design and the second involves estimating the accuracy of the metamodel in estimating the response at a new set of untried design variables. In this paper, we focus on the second method.

The most commonly used method in statistical regression modelling is finding the coefficient of determination, R^2 . However, the Kriging model characteristically fits the training design exactly, therefore, calculating R^2 will always produce a value of 1. An alternative method is to run the simulation again using new design variables sets, use the metamodel to predict the response at these new locations and then estimate the error. However, in cases where running the simulation is time-consuming, this method is not recommended. Where the Kriging model is used, the cross-validation root mean squared error (CV-RMSE) can be estimated (Meckesheimer et al., 2002).

Cross-validation (CV) groups the training design into a total of n_f “folds” containing M/n_f training sets. Each fold is used in both training and testing the metamodel. For example, for a training design with 25 design variable sets, if a “leave-one-out” CV method is used then we have $n_f = M$ folds each containing 1 training set. When one design variable set is withheld, the other 24 design variable sets are used to estimate the parameters of the chosen metamodel. Using this newly fit metamodel, an estimate of the response at the withheld design variable set is obtained. This process is repeated until all training sets are used in fitting and response estimation. In the end, there are now 25 estimated responses that can be used with their values from the computer simulation to estimate the root mean squared error (RMSE). The RMSE calculated in this way, using CV is called the Cross Validation Root Mean Squared Error (CV-RMSE).

$$CV - RMSE = \sqrt{\frac{\sum_{i=1}^M (z_i - \hat{z}_i)^2}{M}} \quad \dots \text{Eq.23}$$

4. First Order Reliability Method (FORM)

Parameter design attempts to find the distribution parameters, of random design variables, that minimise system failure. To achieve this, there are two main points that must be addressed. First, how do we relate the response to the critical limits to determine when ‘success’ or ‘failure’ occurs and second, how do we calculate the respective probabilities (Madsen et al., 1986; Melchers, 1987). The first is answered using a ‘limit state function’ (LSF) and using the LSF the failure probability is estimated using FORM.

When provided with an upper or lower specification (demand), ζ , the limit-state function (LSF) is written as

$$g(\mathbf{v}) = \pm(\zeta - y(\mathbf{v})) \quad \dots \text{Eq.24}$$

where $y(\mathbf{v})$ is the metamodel. By definition,

$$\begin{aligned} g(\mathbf{v}) = 0 & \quad \mathbf{v} \in \text{Limit - State Surface} \\ g(\mathbf{v}) > 0 & \quad \mathbf{v} \in \text{Success (Conformance)} \\ g(\mathbf{v}) < 0 & \quad \mathbf{v} \in \text{Failure (Non - Conformance)} \end{aligned} \quad \dots \text{Eq.25}$$

The number of LSFs depends upon the number of design specifications.

The probability of failure is normally calculated using the joint probability distribution of the design variables. However, this becomes difficult for cases of multi-dimensional correlated variables (Madsen et al., 1986). In order to simplify this calculation, the LSF is transformed into \mathbf{u} -space where variables follow the standard normal distribution and the probability density function is symmetric. The probability of failure is estimated using FORM. The method makes the assumption that the failure surface can be fitted exactly with a tangent hyper-plane through the Most Likely Failure Point (MLFP), \mathbf{u}^* (Madsen et al., 1986; Melchers, 1987).

Consider the case of two variables shown in Figure 2. Using a suitable transformation, such as the Rosenblatt transformation (Madsen et al., 1986), the variables are transformed from \mathbf{v} -space to \mathbf{u} -space where probability density function is now rotationally symmetric.

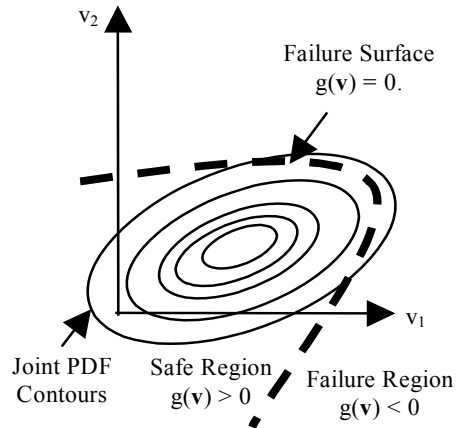


Figure 2. The limit-state surface in \mathbf{v} -space

The linearized failure surface can be rotated so that it is perpendicular to u_1 and parallel to u_2 . Figure 3(a) shows a non-linear LSS, in \mathbf{u} -space, fitted by a hyperplane tangent to the LSS at the MLFP which is also tangent to a normal PDF contour, whereas Figure 3(b) shows the hyperplane rotated to a point perpendicular to the u_1 axis and parallel to u_2 . Notice the position of the new, but equivalent, non-conformance region.

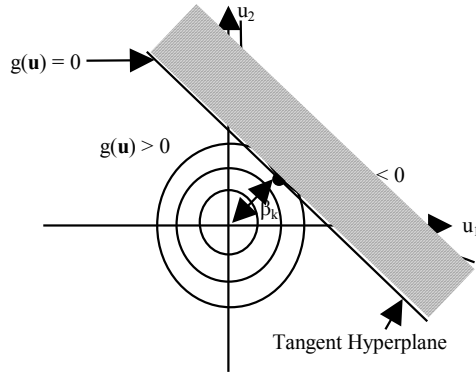


Figure 3(a). A non-linear LSS in u-space

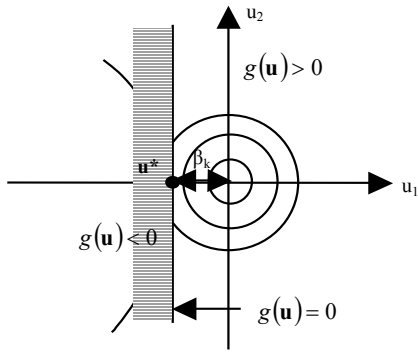


Figure 3(b). The hyperplane rotated

For the bivariate, standard normal density function, the probability of failure is now calculated using Equation 26.

$$\Pr\{F\} = \int_{-\infty}^{-\beta_R} \frac{e^{-\frac{u_1^2}{2}}}{\sqrt{2\pi}} \int_{-\infty}^{\frac{u_2}{2}} \frac{e^{-\frac{u_2^2}{2}}}{\sqrt{2\pi}} du_2 du_1 \quad \dots \text{Eq.26}$$

However, $\int_{-\infty}^{\frac{u_2}{2}} \frac{e^{-\frac{u_2^2}{2}}}{\sqrt{2\pi}} du_2 = 1$, therefore, Equation 26 can be simplified to

$$\Pr\{F\} = \int_{-\infty}^{-\beta_R} \frac{e^{-\frac{u_1^2}{2}}}{\sqrt{2\pi}} du_1 \quad \dots \text{Eq.27}$$

By definition, Equation 27 is the normal cumulative density function (CDF) (Madsen et al., 1986). In general, for the k^{th} LSS,

$$\Pr\{F_k\} = \Pr(g_k(\mathbf{u}) < 0) = \Phi(-\beta_{R_k}) \quad \dots \text{Eq.28}$$

Since the reliability index, β_R , is just the Euclidean distance from the origin to the MLFP, estimating the MLFP is the major step involved in FORM and is found via the constrained optimisation problem,

$$\min_{\mathbf{u}} \sqrt{(\mathbf{u}^T \mathbf{u})} \quad \dots \text{Eq.29}$$

s.t. $g(\mathbf{u}) = 0$

There exists an algorithm in which Equation 29 can be solved (Madsen et al., 1986). However, instead of doing a sub-optimisation routine for every new set of design parameters, another more versatile approach is to find a plane perpendicular (via the null space vectors) to the outward normal gradient vector (Seshadri, 2002). At the MLFP, the vector \mathbf{u}^* is then perpendicular to each of the null space vectors. More specifically, if we have n design variables, there will be $n - 1$ null space vectors each of length n denoted as the matrix $null(\nabla g_k(\mathbf{u}))$.

Then at \mathbf{u}^* , we have the orthogonal conditions as $n - 1$ scalar products

$$\mathbf{u}_k \cdot null(\nabla g_k(\mathbf{u}))_j = 0 \quad j = 1, 2, \dots, n - 1 \quad \dots \text{Eq.30}$$

where \mathbf{u}_k is the MLFP associated with the k^{th} LSF.

5. Searching for a Balanced Design

Probability is associated with β_R , therefore, for an ideal situation, we wish to maximise β_{R_k} simultaneously while keeping all β_{R_k} at about the same value. This is termed a ‘balanced design’. In order to achieve this, suppose a weight is written for each LSF

$$W_k = \exp(-\alpha_k \beta_{R_k}) \quad \dots \text{Eq.31}$$

where each term, α_k , is the sign of $g_k(\mathbf{u}) = 0$. W_i approaches zero for a large, positive product ($\alpha_k \beta_{R_k}$) but becomes large for negative products ($\alpha_k \beta_{R_k}$). In addition, for small changes about a small β_{R_k} , the exponential changes rapidly. A vector of all n_{LF} limit-state functions is defined as

$$\gamma_{LF}^T = [W_1 \dots W_{N_{LF}}] \quad \dots \text{Eq.32}$$

By invoking a sum-of-squares type formulation, we have the objective function

$$Q(\mu_v) = \frac{1}{2} \gamma_{LF}^T \gamma_{LF} = \frac{1}{2} \left[\sum_{k=1}^{N_{LF}} \exp(-2\alpha_k \beta_{R_k}) \right] \quad \dots \text{Eq.33}$$

The objective function of Equation 33 has some interesting properties. If all $\alpha_k = +1$, then the success region contains the origin in \mathbf{u} -space, and the objective function has a lower bound of zero for very large β_R 's and an upper bound of N_{LF} for β_R 's approaching zero. For any $\alpha_k = -1$, such that the success region does not contain the origin, then the objective function is always larger than the number of negative α_k 's.

In summary, the method attempts to move the parameters, $\mathbf{p} = [\mu_1, \mu_2, \dots, \mu_n]$ so that the origin in \mathbf{u} -space is the geometrical centre of the success region. If this is so, for multiple LSSs, then all LSSs are approximately equal distance from the origin and a ‘balanced design’ is achieved. Therefore, to find a balanced design, the optimisation problem of Equation 33 becomes

$$\begin{aligned} & \min_{\boldsymbol{\mu}} \sum_{k=1}^{N_{LF}} \exp(-2\alpha_k \beta_{R_k}) \\ & \text{s.t.} \quad \dots \text{Eq.34} \\ & \mathbf{g}_k(\mathbf{u}) = 0 \\ & \mathbf{u}_k \cdot \text{null}(\nabla \mathbf{g}_k(\mathbf{u}))_j = 0 \\ & LB \leq \mu_k \leq UB \end{aligned}$$

In Equation 34, N_{LF} represents the total number of LSFs, and LB and UB represent the upper and lower boundary conditions on the mean of each design variable. The subscripts k and j are counters for the total number of LSFs, and null space vectors respectively; $k = 1, \dots, N_{LF}$ and $j = 1, \dots, n-l$.

System failure is estimated by first determining whether the components follow a series or parallel arrangement. In this work, we assume the LSFs follow a series arrangement, therefore, if one LSF does not meet specifications, the entire system has “failed”. Assuming that LSFs are uncorrelated and that their intersections are negligible,

$$\Pr(\mathbf{F}) = \sum_{i=1}^{N_{LF}} \Pr(F_i) \quad \dots \text{Eq.35}$$

6. Examples

Three examples are presented with static responses and uncertain variables. Each example contains specific design specifications and our goal is to search for the mean, of each design variable, that allows the system to meet stated design specifications.

6.1 Example 1: Servo System

This simple example is used to illustrate the methodology proposed in figure 1 and appears in (Savage 2011). Consider an electrical-mechanical servo system whose steady-state shaft speed has the function,

$$z(\mathbf{v}) = G \frac{v_2}{v_1} \quad \dots \text{Eq.36}$$

where $G = 2$. In this mechanistic model, there are two variables of interest, v_1 and v_2 . For an optimal design, the mean of each design variable falls within the ranges shown in Equation 37 and the standard deviations are estimated using Equation 38.

$$0.0078 \leq \mu_1 \leq 0.0081 \quad 0.9 \leq \mu_2 \leq 1.1 \quad \dots \text{Eq.37}$$

$$\sigma_1 = \frac{2\mu_1}{300} \quad \sigma_2 = \frac{1.5\mu_2}{300} \quad \dots \text{Eq.38}$$

STEP 1: Building the computer model

Using MATLAB®, a computer simulation based upon the mechanistic model of Equation 36 is developed that allows us to obtain the training responses.

STEP 2: Selecting the training design

In order to generate the training design, three levels of each design variable are used to produce a total of 9 training sets (see Table 1).

Table 1. The three levels of each design variable used to develop the training design

Level	v_1	v_2
Low	0.0078	0.90
Nominal	0.00795	1.0
High	0.0081	1.1

These levels are obtained using the specified ranges of the means of the design variables. Using a full 3^2 factorial design, the training sets are shown in Equation 39,

$$\mathbf{X}_d = \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ \vdots & \vdots \\ x_{9,1} & x_{9,2} \end{bmatrix} = \begin{bmatrix} 0.0078 & 0.90 \\ 0.0078 & 1.0 \\ \vdots & \vdots \\ 0.0081 & 1.1 \end{bmatrix}_{9 \times 2} \quad \dots \text{Eq.39}$$

Using the MATLAB m-file that has been developed based upon the mechanistic model, the corresponding training responses have been obtained using the 9 combinations of the training sets,

$$\mathbf{z} = [230.77 \quad 256.41 \quad 282.05 \quad \dots \quad 271.60]^T \quad \dots \text{Eq.40}$$

STEP 3: Choose a metamodel and estimate model parameters

In this work, we investigate the feasibility of using the RSM and Kriging metamodels. Therefore, the parameters of each model are estimated. For the regression model, a hypothesis test of the model parameters (Montgomery 2005) found that the model parameter corresponding to the v_2^2 term is not significant and can be ignored. Therefore, to put the model in the form $y = \mathbf{w}^T \mathbf{r}(\mathbf{v})$, we have

$$\mathbf{w}^T = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4]_{1 \times 5} \text{ and}$$

$$\mathbf{r}(\mathbf{v}) = [1 \quad v_1 \quad v_2 \quad v_1^2 \quad v_1 v_2]_{1 \times 5}^T.$$

For the universal kriging model, we have

$$\mathbf{w}^T = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \Gamma^{-1}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta})] \text{ and}$$

$$\mathbf{r}^T(\mathbf{v}) = [1 \quad v_1 \quad v_2 \quad \tilde{\mathbf{r}}^T(\mathbf{v})]$$

where \mathbf{z} is the vector of training responses shown in Equation 40,

$$\mathbf{X} = \begin{bmatrix} 1 & 0.0078 & 0.9 \\ \vdots & \vdots & \vdots \\ 1 & 0.0081 & 1.1 \end{bmatrix}_{9 \times 3} \text{ and}$$

$$\tilde{\mathbf{r}}(\mathbf{v}) = \begin{bmatrix} \exp(-\theta_1(v_1 - 0.0078)^2 - \theta_2(v_2 - 0.9)^2) \\ \exp(-\theta_1(v_1 - 0.0078)^2 - \theta_2(v_2 - 1.0)^2) \\ \vdots \\ \exp(-\theta_1(v_1 - 0.0081)^2 - \theta_2(v_2 - 1.1)^2) \end{bmatrix}_{9 \times 1}$$

The next step is to check the fit of the metamodels which is done using the cross-validation method.

STEP 4: Checking the fit of the metamodel

Since our training design consists of a total of only 9 training sets, a “leave-one-out CV” method is used to calculate the root mean squared error. These results for each metamodel are shown in Table 2.

Since CV-RMSE estimates the prediction error, a smaller value indicates a better fit. Although both metamodels produced very small and, thus, acceptable results, the RSM produced a smaller CV-RMSE than Kriging. These results indicate that the RSM provided a better fit than the Kriging model.

Table 2. Comparison of metamodel performance using the CV-RMSE

Model	CV-RMSE
RSM	1.204 x 10 ⁻²
Kriging	3.159 x 10 ⁻²

STEP 5: Building the LSFs

This problem requires a ‘target-is-best’ solution and we are given upper and lower design specifications of 255 rad/s and 245 rad/s respectively. Following Equation 24, the limit-state functions look like

$$g_1(\mathbf{v}) = 255 - y(\mathbf{v}) \quad \dots \text{Eq.41}$$

$$g_2(\mathbf{v}) = y(\mathbf{v}) - 245 \quad \dots \text{Eq.42}$$

STEP 6: Transform the LSFs from v-space to u-space

The LSFs are now transformed to **u**-space using a transformation method like the Rosenblatt transformation (Madsen et al., 1986). Since the variables are normally distributed and are uncorrelated, to

transform the LSF to **u**-space, the following transformation is used.

$$v_i = \mu_i + \sigma_i u_i \quad \dots \text{Eq.43}$$

After substituting equation 43 into Equations 41 and 42, the LSF is now transformed to **u**-space.

STEP 7: Search for the best design

For this problem, we focus on finding the design parameters, **p** = [μ₁ μ₂], that produce a minimum system failure. In order to first find a feasible design, our optimisation problem looks like

$$\min_{\mu} Q = \exp(-2\alpha_1 \beta_{R_1}) + \exp(-2\alpha_2 \beta_{R_2}) \quad \dots \text{Eq.44}$$

Subject to :

$$\begin{aligned} \mathbf{u}_1 \cdot \text{null}(\nabla g_1(\mathbf{u}))_1 &= 0 \\ \mathbf{u}_2 \cdot \text{null}(\nabla g_2(\mathbf{u}))_1 &= 0 \\ 0.0078 \leq \mu_1 \leq 0.0081 \\ 0.9 \leq \mu_2 \leq 1.1 \end{aligned}$$

Since this problem consists of 2 design variables, then there is only 1 null-space vector $\text{null}(\nabla g_k(\mathbf{u}))_1$. The results of the optimisation using the mechanistic model and both metamodels, with FORM, will be presented. These results are shown in Table 3. Our design already produces failure probabilities that are acceptable (< 0.01).

In Table 3, the second column shows the best design found using the corresponding model, identified in column 1. The MLFP at the best design is shown where **u**₁^{*} corresponds to the MLFP of the first LSS (whose function is equation 41) and **u**₂^{*} is the MLFP at the second LSS (whose function is Equation 42).

Table 3. The best design calculated using FORM with the different models

Model	Best Design	MLFP	β _{R₁}	β _{R₂}	
Mechanistic	$\mathbf{p} = \begin{bmatrix} 0.008001 \\ 0.9997 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} -1.944 \\ 1.429 \end{bmatrix}$	$\mathbf{u}_2^* = \begin{bmatrix} 1.896 \\ -1.450 \end{bmatrix}$	2.413	2.387
RSM	$\mathbf{p} = \begin{bmatrix} 0.008001 \\ 0.9999 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} -1.929 \\ 1.418 \end{bmatrix}$	$\mathbf{u}_2^* = \begin{bmatrix} 1.907 \\ -1.460 \end{bmatrix}$	2.394	2.402
Kriging	$\mathbf{p} = \begin{bmatrix} 0.008001 \\ 0.9999 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} -1.929 \\ 1.419 \end{bmatrix}$	$\mathbf{u}_2^* = \begin{bmatrix} 1.910 \\ -1.462 \end{bmatrix}$	2.395	2.405

The reliability indexes, β_{R₁} and β_{R₂}, of the first and second LSSs are shown. The results from the mechanistic model are assumed to be the “true” or “most accurate” results. The difference in the failure points obtained from the metamodels and the mechanistic models is very small (< 0.1%).

In order to visualise these results, the limit-state surfaces are plotted along with the normal PDF contours and shown in Figure 4. In this work, the normal PDF numbers just show that the LSSs are about the same distance away from the origin, therefore resulting in a

“balanced design”. This is also observed in Table 3 where β_{R₁} and β_{R₂} do not differ by much. Now, in order to test the reliability of FORM, a MCS is run to check if the probabilities are consistent.

On observing Figure 4, the LSSs are straight lines. Therefore, their tangent hyperplane approximations are, theoretically exact. As a result, FORM should provide an exact estimate of the failure probability. In order to check this assumption, a MCS using each model is done and the failure probability obtained is compared with the FORM failure probability estimate. These results are

shown in Table 4.

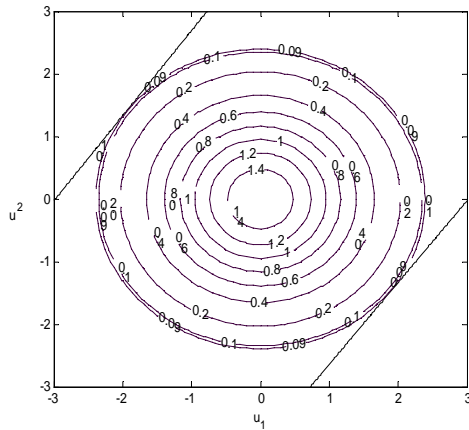


Figure 4. The two limit-state surfaces (in u-space) and the joint Normal PDF contours

As shown in Table 4, $\Pr(F_1) \equiv \Pr(g_1(\mathbf{v}) < 0)$ and $\Pr(F_2) \equiv \Pr(g_2(\mathbf{v}) < 0)$. The number of samples required for the MCS, depends upon the order of the probability of failure to be estimated (Andrieau-Renaud et al., 2004). To accurately compute a probability of failure of 10^{-n} , typically, somewhere between 10^{n+2} and 10^{n+3} samples are required. Therefore, for this problem, somewhere between 10^5 and 10^6 samples are needed. Here, we use 500,000 samples.

There are differences in the probability of failure estimated but in the worst case, this difference is 5% and is the result of the difference in the failure points at each model. The computation time shown is the time required to estimate the failure probability at the best design. Kriging requires the longest time because of the additional effort required to compute the Kriging model derivatives. To check the accuracy of FORM, a Monte Carlo Simulation of each model is done.

Table 4. The probabilities of failure estimated using FORM and a Monte Carlo Simulation with each model

Model	Best Design	FORM			MCS (500,000 runs)		
		Pr(F ₁)	Pr(F ₂)	CPU Time	Pr(F ₁)	Pr(F ₂)	CPU Time
Mechanistic	$\mathbf{p} = \begin{bmatrix} 0.008000 \\ 0.9997 \end{bmatrix}$	0.007912	0.008490	1.7s	0.007778	0.008606	< 1s
RSM	$\mathbf{p} = \begin{bmatrix} 0.008001 \\ 0.9999 \end{bmatrix}$	0.008323	0.008159	1.9s	0.008505	0.008030	< 1s
Kriging	$\mathbf{p} = \begin{bmatrix} 0.008001 \\ 0.9999 \end{bmatrix}$	0.008285	0.008133	~3s	0.008404	0.008118	~4.5s

FORM computes a probability that is consistent with the MCS failure probability. In the worst case, this difference is approximately 2%. Due to the nature of the Kriging model, being more complex than the RSM, or in this case, also the mechanistic model, a MCS of the Kriging model takes a longer time. The speed of the RSM, however, is almost on par to the speed of the simple mechanistic model.

6.2 Example 2: Thin Film Layer

This example is adopted from (Bagchi and Templeton 1994). The electrical impedance of a film is given explicitly as a function of three variables v_1 , insulator thickness, v_2 , conductor line width and v_3 , line height, through the equation.

$$z(\mathbf{v}) = \frac{87}{\sqrt{\varepsilon + \sqrt{2}}} \ln \left(\frac{5.98v_1}{0.8v_2 + v_3} \right) \quad \dots \text{Eq.45}$$

where the dielectric constant of the material $\varepsilon = 3.094539$.

STEP 1: Building the computer model

A computer model based upon the mechanistic model,

Equation 45, is built using MATLAB[®].

STEP 2: Selecting the training design

Using the distribution of each, $v_1 \sim N(26.6, 0.3333^2)$, $v_2 \sim N(17.5, 0.2222^2)$ and $v_3 \sim N(6.0, 1.1111^2)$, three levels of each variable are obtained (see Table 5) and a full 3^3 factorial training design is developed.

Table 5. The three levels of each design variable used to develop the training design

Level	v_1	v_2	v_3
Low	24.6	16.16	5.34
Nominal	26.6	17.5	6.00
High	28.6	18.84	6.67

STEPS 3 and 4: Choose a metamodel and estimate parameters and check the fit of the metamodel

As in the previous example, the parameters of both the RSM and universal Kriging models are estimated and their fit is checked using CV to estimate the RMSE.

Table 6 depicts these results. It is shown that the RSM provided a better fit of the training design than the Kriging model.

Table 6. The CV-RMSE estimate of each Metamodel

Model	CV-RMSE
RSM	5.469×10^{-4}
Kriging	2.489×10^{-3}

STEP 5: Building the LSF

In this problem, the upper and lower specification limits of 86 and 84 produce the limit state functions.

$$g_1(\mathbf{v}) = 86 - y(\mathbf{v}) \quad \dots \text{Eq.46}$$

$$g_2(\mathbf{v}) = y(\mathbf{v}) - 84 \quad \dots \text{Eq.47}$$

STEP 6: Transform variables from v-space to u-space

The Rosenblatt transformation is normally used to transform the variables from v-space to normal, uncorrelated u-space. A detailed description of this transformation is provided in (Madsen et al., 1986). Since the variables are normally distributed and uncorrelated, the transformation function looks like

$$v_i = \mu_i + \sigma_i u_i \quad \dots \text{Eq.48}$$

STEP 7: Searching for the best design

The optimisation looks like (44) with new bounds on the means; $[25.5 \ 16.83 \ 5.67]^T \leq \mu_v \leq [27.6 \ 18.17 \ 6.33]^T$. Since this problem consists of 3 design variables, then are two null-space vectors; $null(\nabla g_k(\mathbf{u}))_1$, $null(\nabla g_k(\mathbf{u}))_2$. The results of the optimisation using the mechanistic model and both metamodels with FORM, are presented.

Although the better fit of the RSM produces a closer design when compared with the results from the mechanistic model (see Table 7). This is observed in both the location of the MLFP and the estimation of the reliability index. This difference is not very great since the worse difference being approximately 1%. The difference between reliability indexes is less than 0.2%. The failure probabilities at the best design, estimate using both FORM and a MCS with 100,000 runs, are shown in Table 8.

Similar to that of the previous example, $\Pr(F_1) \equiv (86 - y(\mathbf{v})) < 0$ and $\Pr(F_2) \equiv (y(\mathbf{v}) - 84) < 0$. In this example, Kriging seems to be more conservative than the RSM since the failure probability estimate is higher than the mechanistic model-based FORM results.

Table 7. The best design found using each metamodel with FORM

Model	Best Design		MLFP	β_{R_1}	β_{R_2}
Mechanistic	$\mathbf{p} = \begin{bmatrix} 27.60 \\ 18.00 \\ 6.329 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} 1.184 \\ -0.8643 \\ -0.5211 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} -1.190 \\ 0.8299 \\ 0.5229 \end{bmatrix}$	1.556	1.542
RSM	$\mathbf{p} = \begin{bmatrix} 27.60 \\ 18.01 \\ 6.328 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} 1.185 \\ -0.8665 \\ -0.5219 \end{bmatrix}$	$\mathbf{u}_2^* = \begin{bmatrix} -1.191 \\ 0.8295 \\ 0.5226 \end{bmatrix}$	1.558	1.543
Kriging	$\mathbf{p} = \begin{bmatrix} 27.60 \\ 18.00 \\ 6.330 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} 1.183 \\ -0.8519 \\ -0.5244 \end{bmatrix}$	$\mathbf{u}_2^* = \begin{bmatrix} -1.188 \\ 0.8319 \\ 0.5198 \end{bmatrix}$	1.553	1.544

Table 8. Probability of failure estimated using a MCS of the Metamodel at the best design

Model	Best Design	FORM			MCS (100,000 runs)		
		$\Pr(F_1)$	$\Pr(F_2)$	CPU Time	$\Pr(F_1)$	$\Pr(F_2)$	CPU Time
Mechanistic	$\mathbf{p} = \begin{bmatrix} 27.60 \\ 18.01 \\ 6.329 \end{bmatrix}$	0.05986	0.06158	~1.5s	0.05957	0.06097	< 1s
RSM	$\mathbf{p} = \begin{bmatrix} 27.60 \\ 18.01 \\ 6.328 \end{bmatrix}$	0.05965	0.06147	~1.5s	0.05865	0.06114	< 1s
Kriging	$\mathbf{p} = \begin{bmatrix} 27.60 \\ 18.00 \\ 6.330 \end{bmatrix}$	0.06070	0.06172	~6s	0.06024	0.06166	~1.5s

The difference in the probabilities is acceptable since the worse case difference ~ 2%. Like the previous example, the Kriging-based FORM method is slower than the RSM-based method. FORM is again consistent with Monte Carlo results.

6.3 Example 3: Nonlinear LSF

Consider the following mathematical problem, shown in Equation 48, based upon a problem from (Youn and Choi, 2004).

$$\min_{\mu} f = (\mu_1 + \mu_2)$$

Subject to:

$$\Pr\left[g_1 = \frac{v_1^2 v_2}{20} - 1 \geq 0\right] \leq \Phi(-\beta_{R_1}) \quad \dots \text{Eq.49}$$

$$\Pr\left[g_2 = \left(\frac{(v_1 + v_2 - 5)^2}{30} + \frac{(v_1 - v_2 - 12)^2}{120}\right) - 1 \geq 0\right] \leq \Phi(-\beta_{R_2})$$

$$2.75 \leq \beta_{R_i} \leq 3.00$$

$$\begin{bmatrix} 1.2 \\ 0.2 \end{bmatrix} \leq \mu_v \leq \begin{bmatrix} 4.8 \\ 3.8 \end{bmatrix}$$

It is assumed that each variable has the distribution $v_i \sim N(\mu_i, 0.3^2)$. Using the means $\mu_1 = 3$ and $\mu_2 = 2$ along with the relation $\mu_i \pm 6\sigma$, five levels of each variable, v_1 and v_2 , are obtained. The response of these 25 design variable sets is then generated for each function, g_1 and g_2 . A response surface model (RSM) and Kriging model is then fit and the associated CV-RMSE is estimated to compare the fit of each metamodel.

Table 9 shows the results. Kriging provided a better fit for the first mechanistic model but the RSM provided a better fit for the second. The fit of the Kriging metamodel is consistent since the both errors are in the order of 10^{-5} . This is not true for the RSM. One fit is bad and the other is very good.

Table 9. CVRMSE of each metamodel for each LSF

Level	g_1	g_2
RSM	0.0276	1.742×10^{-15}
Kriging	1.082×10^{-5}	1.718×10^{-5}

FORM is used with each metamodel to calculate the best design. This problem has two steps to completion; the first involves finding the feasible design using Equation 34 and the second step involves optimising Equation 49. The best design is also calculated using the original functions with FORM.

Table 10 depicts the results. It is shown that the best design found using the Kriging metamodel is closer to that found using the mechanistic model, although the RSM design was not bad. As seen in the previous examples, Kriging takes a longer time to find the MLFP than the RSM. Figure 5 shows a plot of the non-linear limit-state surfaces and the normal PDF contours. The solid line corresponds to g_1 and the dotted line corresponds to g_2 .

Using the MATLAB function ‘norminv’ the reliability index is estimated using the probability of failure from the MCS of each model. The results are shown in Table 11. Although not exactly equation to 3.00 and 2.75, the designs produce reliability indexes that fall well within the specified boundaries. In terms of speed, the RSM is still faster than Kriging.

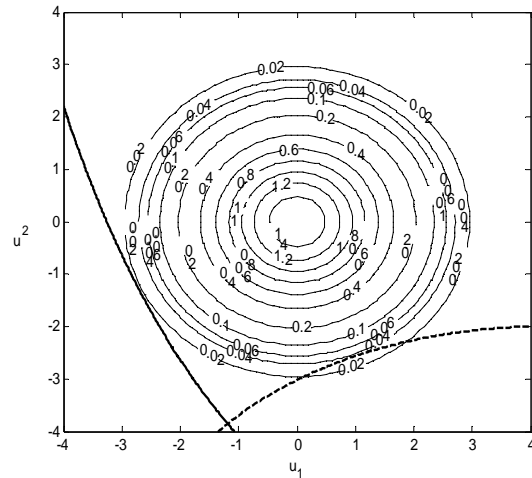


Figure 5. The two limit-state surfaces, in u-space, along with the normal PDF contours

Table 10. The best design found using each model with FORM and the corresponding reliability indexes for each LSF

Model	Best Design	MLFP	β_{R_1}	β_{R_2}	CPU Time	
Mechanistic	$\mathbf{p} = \begin{bmatrix} 3.470 \\ 3.218 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} -2.717 \\ -1.272 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} 0.9706 \\ -2.573 \end{bmatrix}$	3.00	2.75	~ 2s
RSM	$\mathbf{p} = \begin{bmatrix} 3.405 \\ 3.193 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} -2.623 \\ -1.456 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} 1.028 \\ -2.551 \end{bmatrix}$	3.00	2.75	~ 1.5s
Kriging	$\mathbf{p} = \begin{bmatrix} 3.471 \\ 3.218 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} -2.719 \\ -1.269 \end{bmatrix}$	$\mathbf{u}_1^* = \begin{bmatrix} 0.9702 \\ -2.573 \end{bmatrix}$	3.00	2.75	~ 8.5s

Table 11. The probability of failure, at the best design, estimated using a Monte Carlo Simulation (500,000 runs)

Model	Best Design	Pr(F ₁)	Pr(F ₂)	CPU Time
Mechanistic	$\mathbf{p} = \begin{bmatrix} 3.470 \\ 3.218 \end{bmatrix}$	0.001450 (2.978)	0.002446 (2.814)	< 1s
RSM	$\mathbf{p} = \begin{bmatrix} 3.405 \\ 3.193 \end{bmatrix}$	0.001522 (2.963)	0.002530 (2.803)	< 1s
Kriging	$\mathbf{p} = \begin{bmatrix} 3.471 \\ 3.218 \end{bmatrix}$	0.001518 (2.964)	0.002574 (2.798)	~ 8s

8. Conclusion

This paper has presented a simplified approach for parameter design of static systems when provided with random design variables. The method first uses a metamodel to approximate the mechanistic model which is then used to build the LSF based upon the design criteria. Finally, FORM is used to estimate the probability of failure. Since the design variables are random, the method searches for the means of the design variables that result in a reduced failure probability.

Three examples have been presented to illustrate and show the accuracy of this methodology for metamodel-based parameter design. Two different metamodels have been used and the best design found using each metamodel, along with the failure point, reliability index and probability of failure have been compared. Since the examples presented are fairly simple, the failure point, reliability index and failure probability have also been found using the mechanistic model; these were considered to be the most accurate results and used to test the accuracy of the metamodel-based FORM results.

The examples have shown that the metamodels worked very well with FORM to find a design that is very close to the results from the mechanistic model. The accuracy of FORM is very good when compared to the Monte Carlo Simulation showing that FORM is reliable and has potential to replace the MCS to estimate the failure probability. In terms of speed, Kriging was the slower metamodel. Although FORM has been generally slower than a Monte Carlo simulation in estimating probabilities, FORM is able to produce a balanced design without adding additional constraints.

The metamodel-based parameter design methodology has potential and further work involves increasing the complexity of the mechanistic models to include highly nonlinear or implicit mechanistic models. This method would also be applied to the design of systems with a dynamic response.

References:

Allen, M., Raulli, M., Maute, K. and Frangopol, D.M. (2004), "Reliability-based analysis and design optimisation of

- electrostatically actuated MEMS", *Computers and Structures*, Vol.82, pp.1007-1020
- Andrieau-Renaud, C., Sudret, B. and Lemaire, M. (2004), "The PH12 method: a way to compute time-variant reliability", *Reliability Engineering and System Safety*, Vol.84, pp.75-86
- Bagchi, T. and Templeton, J. (1994), "Robust design engineering using variance transfer", *Proceedings of ISSAT 2nd International Conference on Reliability and Quality in Design*, Seattle, March, pp.203-207
- Barton, R.R. (1998), "Simulation metamodels", *Proceedings of the IEEE Winter Simulation Conference 1998*, Washington D.C., USA, December, pp. 167-174
- Deng, J. (2006), "Structural reliability analysis for implicit performance function using radial basis function network", *International Journal of Solids and Structures*, Vol.43, Nos.11/12, pp.3255-3291
- Esfandiari, R.S. and Lu, B. (2010), *Modeling and Analysis of Dynamic Systems*, CRC Press, USA.
- Grandhi, R.V. and Wang, L. (1998), "Reliability-based structural optimisation using improved two-point adaptive nonlinear programming", *Finite Elements in Analysis and Design*, Vol.29, pp.35-48
- Hussain, M.K., Barton, R.R. and Joshi, S.B. (2002), "Metamodeling: radial basis functions versus polynomials", *European Journal of Operational Research*, Vol.138, No.1, pp.142-154
- Jin, R., Chen, W. and Simpson, T.W. (2001), "Comparative studies of metamodeling techniques under multiple modelling criteria", *Structural Multidisciplinary Optimisation*, Vol.23, No.1, pp.1-13
- Lobontiu, N. (2010), *System Dynamics for Engineering Students: Concepts and Applications*, Elsevier Inc., USA.
- Madsen, H.O., Krenk, S. and Lind, N.C. (1986), *Methods of Structural Safety*, Prentice Hall Inc. USA
- Martin, J.D. and Simpson, T.W. (2005), "Use of Kriging models to approximate deterministic computer models", *AIAA Journal*, Vol.43, No.4, pp.853-863
- Meckesheimer, M., Booker, A.J., Barton, R.R. and Simpson, T.W. (2002), "Computationally inexpensive metamodel assessment strategies", *AIAA Journal*, Vol. 40, No.10, pp.2053- 2060
- Melchers, R.E. (1987), *Structural Reliability: Analysis and Prediction*, John Wiley and Sons, England.
- Montgomery, D.C. (2005), *Design and Analysis of Experiments*, John Wiley and Sons Inc., USA
- Muller, A.A. and Messac, A. (2006), "Metamodeling using extended radial basis functions: a comparative approach", *Engineering with Computers*, Vol.21, pp.203-217
- Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989), "Design and analysis of computer experiments", *Statistical Science*, Vol.4, No.4, pp.409-423
- Savage, G.J. (2011), *Model-based Robust Design*, Course Notes (Systems Models 2: SYDE 554), Department of Systems Design Engineering, University of Waterloo, Waterloo, Canada
- Seshadri, R. (2002), *Integrated Robust Design Using Probability of Conformance Metrics*, Thesis (unpublished), University of Waterloo, Canada
- Schueremans, L. and van Gemert, D. (2005), "Benefit of splines and neural networks in simulation-based structural reliability analysis", *Structural Safety*, Vol.27, No.3, pp.246-261
- Simpson, T.W., Mauery, T.M., Korte, J.J. and Mistree, F. (2001), "Kriging models for global approximation in simulation-based multidisciplinary design optimisation", *AIAA Journal*, Vol.39, No.12, pp.2233-2241
- Youn, B.Y. and Choi, K.K. (2004), "Selecting probabilistic approaches for reliability-based design optimisation", *AIAA Journal*, Vol.42, No.1, pp.124-131

Appendix A:**Converting the RSM to matrix-vector form**

Consider a 2nd order RSM with two design variables as shown in equation.

$$y(\mathbf{v}) = \beta_0 + \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_1^2 + \beta_4 v_2^2 + \beta_5 v_1 v_2 \quad (\text{A.1})$$

$$\mathbf{X}_a = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,1}^2 & x_{1,2}^2 & x_{1,1}x_{1,2} \\ 1 & x_{2,1} & x_{2,2} & x_{2,1}^2 & x_{2,2}^2 & x_{2,1}x_{2,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{M,1} & x_{M,2} & x_{M,1}^2 & x_{M,2}^2 & x_{M,1}x_{M,2} \end{bmatrix}_{M \times 6} \quad (\text{A.2})$$

Grouping and separating the constants, $\boldsymbol{\beta}$ and the variables, \mathbf{v} we can get two vectors, (A.3) and (A.4), that when multiplied, according to equation (1) produces the exact form of (A.1).

$$\mathbf{w}^T = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \beta_3 \quad \beta_4 \quad \beta_5] \quad (\text{A.3})$$

$$\mathbf{r}(\mathbf{v}) = [1 \quad v_1 \quad v_2 \quad v_1^2 \quad v_2^2 \quad v_1 v_2]^T \quad (\text{A.4})$$

Appendix B:**Converting the Kriging model to matrix-vector form**

Consider the universal kriging model consisting of 2 design variables, v_1 and v_2 . In its common form, this model would look like

$$y(\mathbf{v}) = \beta_0 + \beta_1 v_1 + \beta_2 v_2 + \tilde{\mathbf{r}}^T(\mathbf{v}) \boldsymbol{\Gamma}^{-1}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}) \quad (\text{B.1})$$

where

$$\tilde{\mathbf{r}}^T(\mathbf{v}) = \left[\exp\left(-\sum_{i=1}^2 (\theta_i (v_i - x_{i,1})^2)\right) \quad \dots \quad \exp\left(-\sum_{i=1}^2 (\theta_i (v_i - x_{M,i})^2)\right) \right]_{1 \times M} \quad (\text{B.2})$$

Equation (B.2) is simplified to look like equation (B.3) where each element of ρ looks like (B.4).

$$\tilde{\mathbf{r}}^T(\mathbf{v}) = [\rho_1(\mathbf{v}) \quad \dots \quad \rho_M(\mathbf{v})]_{1 \times M} \quad (\text{B.3})$$

$$\rho_j(\mathbf{v}) = \theta_1 (v_1 - x_{j,1})^2 + \theta_2 (v_2 - x_{j,2})^2 \quad (\text{B.4})$$

Grouping and separating constants and variables, we

$$\tilde{\boldsymbol{\beta}}^T = [\beta_1 \quad \beta_2], \quad \mathbf{f}(\mathbf{v}) = [v_1 \quad v_2]^T, \quad \boldsymbol{\beta} = [\beta_0 \quad \beta_1 \quad \beta_2]^T \text{ and}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} \\ 1 & x_{2,1} & x_{2,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{M,1} & x_{M,2} \end{bmatrix}_{M \times 3}$$

From these groupings, $\tilde{\boldsymbol{\beta}}^T$, $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}^{-1}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta})$ are all vectors whose elements contain constants. Therefore, we can further group all these terms into the matrix \mathbf{w}^T , where, for our 2 design variables, universal kriging model looks like.

$$\mathbf{w}^T = \left[\beta_0 \quad \beta_1 \quad \beta_2 \quad \left(\boldsymbol{\Gamma}^{-1}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}) \right)^T \right]_{1 \times (M+3)} \quad (\text{B.5})$$

$$\mathbf{r}(\mathbf{v}) = [1 \quad v_1 \quad v_2 \quad \tilde{\mathbf{r}}^T(\mathbf{v})]^T_{(M+3) \times 1} \quad (\text{B.6})$$

Appendix C:**Differentiating the Kriging Model**

Following Equation 2, differentiating the Kriging model involves differentiating (B.6). Therefore, the first derivative with respect to v_1 looks like.

$$\frac{\partial \mathbf{r}(\mathbf{v})}{\partial v_1} = \begin{bmatrix} 0 & 1 & 0 & \left(\frac{\partial \tilde{\mathbf{r}}(\mathbf{v})}{\partial v_1} \right)^T \end{bmatrix}^T \quad (\text{C.1})$$

Now, using equations (B.3) and (B.4),

$$\frac{\partial \tilde{r}_j(\mathbf{v})}{\partial \rho_j(\mathbf{v})} = -\exp(-\rho_j(\mathbf{v})) \quad (\text{C.2})$$

$$\frac{\partial \rho_j(\mathbf{v})}{\partial v_1} = 2\theta_1 (v_1 - x_{j,1}) \quad (\text{C.3})$$

Substituting equations (C.2) and (C.3) into equation (21) we get

$$\frac{\partial \tilde{r}_j(\mathbf{v})}{\partial v_1} = -2\theta_1 (v_1 - x_{j,1}) \exp(-\rho_j(\mathbf{v})) \quad (\text{C.4})$$

But $r_j(\mathbf{v}) = \exp(-\rho_j(\mathbf{v}))$, therefore,

$$\left(\frac{\partial \tilde{\mathbf{r}}(\mathbf{v})}{\partial v_1} \right)^T = [-2\theta_1 (v_1 - x_{1,1}) r_1(\mathbf{v}) \quad \dots \quad -2\theta_1 (v_1 - x_{M,1}) r_M(\mathbf{v})] \quad (\text{C.5})$$

Substituting (C.5) back into (C.1)

$$\frac{\partial \mathbf{r}(\mathbf{v})}{\partial v_1} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -2\theta_1 (v_1 - x_{1,1}) r_1(\mathbf{v}) \\ \vdots \\ -2\theta_1 (v_1 - x_{M,1}) r_M(\mathbf{v}) \end{bmatrix}_{(M+3) \times 1} \quad (\text{C.6})$$

This is repeated for v_2 .

Authors' Biographical Notes:

Turuna S. Seecharan obtained her B.Sc. from The University of the West Indies in 2005 in the field of Industrial Engineering. She then began her postgraduate work, in the department of Systems Design Engineering at The University of Waterloo, Canada, in the area of Modelling, Simulation and Systems Theory under the supervision of Prof. Gordon Savage. After obtaining her M.Sc. she continued her research at The University of Waterloo where she is currently a PhD candidate. Her research involves integrating metamodels and reliability methods for the design of systems where uncertainty is considered in the design variables.

Gordon J. Savage was born in Canada and received his PhD from The University of Waterloo in 1977 in the area of Graph-Theoretic Models of Engineering Systems. He is presently a faculty member of, and a full professor in, Systems Design Engineering at The University of Waterloo, Waterloo, Ontario, Canada. His present research interests are in the areas of a) modelling, formulation and computer implementation of linear graph models of engineering systems, and b) design for quality, reliability and robustness in complex engineering systems. Professor Savage has published over 90 refereed papers. Research is ongoing to integrate quality and reliability metrics using both mechanistic and empirical models. He has been involved in the study of the methodology of design for over 30 years and spent parts of seven years presenting the process of innovation in China. His practical experience in industry and his supervision of the Midnight sun solar car project give him a unique ability to link the methodology of design, the process of innovation and robust design.

■