A Bayesian Approach for Multiple Response Surface Optimization in the Presence of Noise Variables

Guillermo Miró-Quesada, Enrique Del Castillo, and John J. Peterson *

October 1, 2002

Abstract

An approach for the multiple response robust parameter design problem based on a methodology by Peterson (2000) is presented. The approach is Bayesian, and consists of maximizing the posterior predictive probability that the process satisfies a set of constraints on the responses. In order to find a solution robust to variation in the noise variables, the predictive density is integrated not only with respect to the response variables but also with respect to the assumed distribution of the noise variables. The maximization problem involves repeated Monte Carlo integrations, and two different methods to solve it are evaluated. A Matlab code was written that rapidly finds an optimal (robust) solution in case it exists. Two examples taken from the literature are used to illustrate the proposed method.

1 Introduction

Robust Parameter Design (RPD) is a set of techniques that consists in determining the levels of some set of controllable factors that reduce the sensitivity of the process to variations in another set of uncontrollable factors, the noise factors, thus increasing the robustness of the process. The implicit assumption is that noise factors can be controlled only in an R&D environment but not during manufacturing or during actual use of the product. This paper presents a Bayesian approach to the RPD problem when there are multiple responses of interest, possibly correlated.

A general approximation to the RPD problem based on response surface methods was first proposed by Box and Jones (1990) and later elaborated by Myers (1992). It consists in explicitly modelling the noise and controllable factors as part of a response surface model based on a carefully controlled experiment. The noise factors are not controllable in general, and assuming

*Guillermo Miró Quesada is a Ph.D. candidate in the Department of Industrial & Manufacturing Engineering, Penn State University, University Park, PA 16802. Enrique Del Castillo is an Associate Professor, Department of Industrial & Manufacturing Engineering, Penn State University, University Park, PA 16802. John J. Peterson is a Director of Statistics, Statistical Sciences Department, UW281A, GlaxoSmithKline Pharmaceuticals, 709 Swedeland Road, King of Prussia, PA 19406.

they will vary over some known distribution, variance and expectation models can be obtained from the fitted response to provide *dual responses* which then can be optimized to solve Taguchi's RPD problem. Constraints can be added on the controllable factors to avoid extrapolation of the models. This approach is known in the response surface literature as the dual response approach (Vining and Myers, 1990, Del Castillo and Montgomery, 1993, Lucas, 1994, Myers, 1997, Vining, 1990). These approaches have been developed, thus far, for single response processes. Furthermore, Peterson (2000) mentions how these approaches do not take into account the uncertainty in the parameter estimates in a formal probabilistic sense.

When there are multiple responses of interest, several techniques have been proposed in the last few years but these do not address the RPD problem. These techniques can be classified in general terms as a) those that do not consider the correlation between the responses or the uncertainty in the parameter estimates (e..g, Desirability methods, see Harrington, 1965, Derringer and Suich, 1980, Del Castillo et al., 1996); b) consider such correlation but do not consider the uncertainty in the parameter estimates (e.g., Khuri and Conlon, 1981, Pignatiello, 1993); or c) consider the uncertainty in the estimates but not the correlation between the responses (Del Castillo, 1996). In contrast, the Bayesian reliability approach by Peterson (2000) considers the correlations between the responses and the uncertainty in the parameter estimates in a formal way.

The purpose of this paper is to extend the approach proposed by Peterson (2000) for multiple response optimization to the Robust Parameter Design case. For this purpose, two modifications in his approach are studied herein: 1) noise variables are included in the integration of the predictive density; 2) use of non-linear programming algorithms for performing the numerical maximizations are studied.

The remainder of this paper is organized as follows. Section 2 gives a description of the methodology proposed by Peterson (2000). Section 3 discusses the use of NLP algorithms and proposes a method to provide good starting solutions. Section 4 presents the methodology for including the noise variables, and section 5 illustrates the approach with two examples taken from the literature.

2 Bayesian Predictive Approach

Peterson (2000) describes a methodology to perform the simultaneous optimization of multiple response surfaces using a Bayesian predictive density function. The following model is employed under the typical assumptions of multivariate regression:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{U} \tag{1}$$

where $\mathbf{Y}$ is a $N \times q$ matrix containing the observations of the q responses, $\mathbf{X}$ is a $N \times p$ design matrix, $\mathbf{B}$ is a $p \times q$ matrix of parameters and $\mathbf{U}$ is a $N \times q$ matrix such that:

$$\begin{aligned} \boldsymbol{u}_{i.} &\sim \mathbf{N}_q(\mathbf{0}, \boldsymbol{\Sigma}) \qquad \forall i = 1, \ldots, N \\ \boldsymbol{u}_{.j} &\sim \mathbf{N}_N(\mathbf{0}, \sigma_j^2 \mathbf{I}_N) \qquad \forall j = 1, \ldots, q \end{aligned}$$

that is, the rows of $\mathbf{U}$ are allowed to be correlated but each column must contain independent random variables. Under the model in equation (1), any new $q \times 1$ vector of observations $\mathbf{y}$ for a given level of controllable factors $\mathbf{x}$ follows the model:

$$\mathbf{y} = \mathbf{B}'\mathbf{x} + \boldsymbol{\epsilon} \tag{2}$$

where $\mathbf{x}$ is a $p \times 1$ vector containing the regressors which are functions of the k controllable factors $\{x_i\}_{i=1}^k$ at which the prediction is desired and $\boldsymbol{\epsilon}$ is a $q \times 1$ vector that has the same distribution as $\boldsymbol{u}'_{i.}$.

Now, the problem of multi-response optimization consists of choosing the values of the k controllable factors x_i such that $\mathbf{y}$ has certain desired properties. It is often the case that these desired properties are for the vector of responses $\mathbf{y}$ to meet some set of specifications. . Let A denote the region of the response space defined by these specifications. The region A can have an arbitrary form. Then the multi-response optimization can be summarized into a single objective consisting of maximizing the probability of having the vector of responses inside the specification region A, i.e., maximizing the following probability of conformance:

$$\begin{aligned} max \quad & p(\mathbf{x}) = P(\mathbf{y} \in A | \mathbf{x}, data) \\ subject\ to: \quad & \\ & \mathbf{x} \in R \end{aligned} \tag{3}$$

where $\mathbf{y}$ is assumed to follow model (2) and R is the region where the model is valid, which is usually taken as the experimental region defined by the design matrix $\mathbf{X}$.

By using the classical non-informative joint prior for $\mathbf{B}$, and $\boldsymbol{\Sigma}$ for the model in equation (1), the Bayesian predictive density for $\mathbf{y}$ can be obtained in closed form and is given by a multivariate t distribution with ν degrees of freedom (see Press, 1989, pp. 136). This is given by:

$$f(\mathbf{y}|\mathbf{x}, data) = \frac{\Gamma\left(\frac{\nu+q}{2}\right)}{(\pi\nu)^{q/2}\Gamma\left(\frac{\nu}{2}\right)} \sqrt{|\mathbf{H}|} \left\{ 1 + \frac{1}{\nu} \left(\mathbf{y} - \widehat{\mathbf{B}}'\mathbf{x}\right)' \mathbf{H} \left(\mathbf{y} - \widehat{\mathbf{B}}'\mathbf{x}\right) \right\}^{-\frac{\nu+q}{2}} \tag{4}$$

where $\widehat{\mathbf{B}}$ is the Ordinary Least Squares (OLS) estimator of $\mathbf{B}$, $\Gamma(\cdot)$ denotes the gamma function and $\mathbf{H}$ is given by:

$$\mathbf{H} = \left(\frac{\nu}{N-p}\right) \frac{\widehat{\boldsymbol{\Sigma}}^{-1}}{1 + \mathbf{x}' \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{x}}$$

where $\widehat{\mathbf{\Sigma}}$ is the usual estimator of $\mathbf{\Sigma}$ calculated from the residuals of the multivariate regression fit (Press, 1982):

$$\widehat{\mathbf{\Sigma}} = \frac{1}{N-p}\left(\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}\right)'\left(\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}\right)$$

Since the predictive density is parameter free, it is useful for making inferences about future observations without having to substitute any parameter by its estimate as proposed by Chiao and Hamada (2001). Hence, the predictive density considers the uncertainty in the parameter estimates. Notice that the $\mathbf{H}$ matrix considers the *scaled prediction variance*, $1 + \mathbf{x}'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{x}$. This is a highly desirable property since solving (3) will avoid points where the fitted model has poor prediction properties and will introduce the uncertainty in the model parameters in the value of the computed probability obtained with (4).

The expected vector of $\mathbf{y}|(\mathbf{x}, data)$ is clearly:

$$E(\mathbf{y}|\mathbf{x}, data) = \widehat{\mathbf{B}}'\mathbf{x}$$

The covariance matrix of $\mathbf{y}|(\mathbf{x}, data)$ exists if $\nu > 2$ and is given by:

$$\begin{aligned} Var(\mathbf{y}|\mathbf{x}, data) &= \tfrac{\nu}{\nu-2}\mathbf{H}^{-1} \\ &= \tfrac{\nu}{\nu-2}\left(\tfrac{\nu}{N-p}\tfrac{\widehat{\mathbf{\Sigma}}^{-1}}{1+\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}}\right)^{-1} \\ &= \tfrac{N-p}{N-p-q-1}\left(1 + \mathbf{x}'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{x}\right)\widehat{\mathbf{\Sigma}} \end{aligned}$$

Therefore, we have that:

$$\mathbf{y}|\mathbf{x}, data \sim \mathbf{T}_q^{\nu}\left(\widehat{\mathbf{B}}'\mathbf{x}, \frac{\nu}{\nu-2}\mathbf{H}^{-1}\right)$$

To obtain $p\left(\mathbf{x}\right)$ we need to integrate (4) over the region A:

$$p\left(\mathbf{x}\right) = \int_A f(\mathbf{y}|\mathbf{x}, data)\, d\mathbf{y} \tag{5}$$

The integration in (5) must be performed numerically. One way of performing this integration is by Monte Carlo simulation from a multivariate t distribution. Given that $\mathbf{y}|\mathbf{x}, data$ is distributed as a multivariate t, we can simulate values of $\mathbf{y}$ using the following equation (see Johnson, 1987, chapter 6):

$$\begin{aligned} \mathbf{y} &= \mathbf{H}^{-1/2}\mathbf{z}\sqrt{\nu/s} + \widehat{\mathbf{B}}'\mathbf{x} \\ &= \left(\tfrac{\nu\mathbf{V}^{-1}}{1+\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x})}\right)^{-1/2}\mathbf{z}\sqrt{\nu/s} + \widehat{\mathbf{B}}'\mathbf{x} \qquad (6) \\ &= \sqrt{\tfrac{1+\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}}{s}}\ \mathbf{V}^{1/2}\mathbf{z} + \widehat{\mathbf{B}}'\mathbf{x} \end{aligned}$$

where $\mathbf{z}$ is sampled from a $\mathbf{N}_q(\mathbf{0}, \mathbf{I}_q)$ distribution, s is sampled from a chi-square with ν degrees of freedom and $\mathbf{V} = \left(\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}\right)' \left(\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}\right)$. Currently available software is able to simulate both of these variates.

3 Optimization

If we use Monte Carlo simulation, we can only obtain an estimate of $p(\mathbf{x})$, $\widehat{p}(\mathbf{x})$, for a given value of $\mathbf{x}$. Our goal is to find $\mathbf{x}$ such that $p(\mathbf{x})$ is maximized according to (3). We suggest three different methods for performing this optimization. The first two were originally suggested by Peterson (2000).

If the number of controllable factors is small, say less than three, then the first method consists in placing a fine grid over the experimental region R and computing the values of $\widehat{p}(\mathbf{x})$ for every $\mathbf{x}$ on the grid. The maximum $\widehat{p}(\mathbf{x}^*)$ is selected from these values. The main advantage of this approach is that it will not be affected by local optima problems, since in general $\widehat{p}(\mathbf{x})$ will not be a convex function.

When the number of controllable factors is larger than three, using a fine grid may become computationally burdensome. For these cases, a moderate sized grid could be used and a logistic regression model (or *meta-model*) can be fitted to the values of $\widehat{p}(\mathbf{x})$. The fitted model will then be optimized in closed form using derivatives or by numerical methods, mainly depending on the order of the logistic model fitted.

The third method consists in performing a direct search using the $\widehat{p}(\mathbf{x})$ values. However, since the computed values of the integrals are not deterministic given that a Monte Carlo integration method is used, the search procedure should be robust against this uncertainty. One such procedure is the Nelder-Mead simplex technique (Nelder, 1964) which can handle the constraints $\mathbf{x} \in R$ by using penalty functions (Bazaraa et al., 1993). The application of this method is eased by the fact that the objective function, being a probability measure, is bounded between 0 and 1. The following penalty function was used in the examples solved in latter sections:

$$B(\mathbf{x}) = \left(\mathbf{h}^+(\mathbf{x})'(\mathbf{x} - \mathbf{1}) + \mathbf{h}^-(\mathbf{x})'(\mathbf{x} + \mathbf{1})\right)^2 \tag{7}$$

where:

$\mathbf{h}^+(\mathbf{x})$ and $\mathbf{h}^-(\mathbf{x})$ are $\Re^k \Rightarrow \Re^k$ functions defined as:

$$h_i^+(x_i) = \begin{cases} 1 & if\ x_i > 1 \\ 0 & if\ x_i < 1 \end{cases} \qquad \forall i = 1, .., k$$

$$h_i^-(x_i) = \begin{cases} 1 & if\ x_i < -1 \\ 0 & if\ x_i > -1 \end{cases} \qquad \forall i = 1, .., k$$

and $\mathbf{1}$ is a $k \times 1$ vector of ones.

The standard error of $\widehat{p}(\mathbf{x})$, the function evaluations input into the optimization algorithm, is estimated by:

$$\widehat{\sigma}_{\widehat{p}(\mathbf{x})} = \sqrt{\frac{1}{N}\widehat{p}(\mathbf{x})(1-\widehat{p}(\mathbf{x}))} \tag{8}$$

where N is the number of Monte Carlo simulations used to evaluate $\widehat{p}(\mathbf{x})$. Since this standard error decreases with the squared root of N, improving the estimate by increasing the number of simulations can become excessively computationally intensive.

Fixing the seed of the random number generator is a common *variance reduction technique* used in Monte Carlo integration, specially when the interest relies on comparing two different values of $p(\mathbf{x})$, as it is during the search procedures of most NLP algorithms (Rubinstein, 1981, pp. 125). Although this completely eliminates the variability in the values of $\widehat{p}(\mathbf{x})$, it makes all of them dependent on the same stream of random numbers. This introduces a bias in the final solution. The magnitude of this bias, however, can be assessed by repeating the complete optimization step with different seeds.

If the seed of the random number generator is fixed, then the optimization problem becomes deterministic and we can use other type of optimization algorithms that, although highly sensitive to noisy observations, are more efficient than the Nelder-Mead. One technique suitable when the gradient cannot be obtained in closed form is Sequential Quadratic Programming (SQP) (Bazaraa et al., 1993).

In order to start the search at a good location, we propose to use the solution to the following mathematical program as a starting point:

$$\begin{aligned} \textit{min} \quad & \mathbf{x}'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{x} \\ \textit{subject to}: \quad & \\ & \widehat{\mathbf{B}}'\mathbf{x} \in A \\ & \mathbf{x} \in R \end{aligned} \tag{9}$$

The optimization in (9) finds the values of $\mathbf{x} \in R$ for which the expected responses are inside the desirable region A and the scaled predicted variance is minimized. In equation (4) we can see that, for such an $\mathbf{x}$, $|\mathbf{H}|$ will be maximized, giving a high value of the predictive density function, yielding a relative large integrand. Thus we would be starting the optimization of (3) where the probability of conformance is large.

We point out that if (9) is infeasible then there does not exist a value of $\mathbf{x} \in R$ such that the mean response meets the specifications. For this case $p(\mathbf{x})$ will always be less than 0.5 and any solution will be poor.

4 Extension to Noise Variables

Noise variables can be controlled during experimentation but cannot be controlled during a manufacturing process or the actual use of the product. These variables are modelled as random variables with known distributions. Finding the levels of the controllable factors that minimize the variance while attaining some pre-specified expected value for the responses is the principle behind Taguchi's Robust Parameter Design Problem (see Taguchi (1986, 1987) and Taguchi and Wu (1985).

The methodology presented in the previous sections can be easily extended to handle noise variables by treating the noise variables as nuisance variables and averaging over them. Let us separate the k factors affecting the response in k_c controllable factors x_i^c and k_n noise factors x_i^n where $k_n + k_c = k$. We group these factors into two vectors, a $k_c \times 1$ vector $\mathbf{x}_c$ containing the controllable factors and a $k_n \times 1$ vector $\mathbf{x}_n$ containing the noise variables. Notice that $\mathbf{x}$ is written in model form, but $\mathbf{x}_c$ and $\mathbf{x}_n$ are not, so we could write $\mathbf{x} = \boldsymbol{\phi}(\mathbf{x}_c, \mathbf{x}_n)$ and $p(\mathbf{x}) = p(\boldsymbol{\phi}(\mathbf{x}_c, \mathbf{x}_n))$ where the function $\boldsymbol{\phi}(\cdot)$ varies according to the model form. Likewise, if we integrate over the noise factors, we will get $p(\mathbf{g}(\mathbf{x}_c))$ where $\mathbf{g}(\cdot)$ also depends on the model form. With this notation, we propose to solve the following optimization problem:

$$\begin{aligned} & max && P(\mathbf{y} \in A | \mathbf{x}_c, data) = p(\mathbf{g}(\mathbf{x}_c)) \\ subject\ to: & && \\ & && \mathbf{x}_c \in R_c \end{aligned} \tag{10}$$

where :

$$p(\mathbf{g}(\mathbf{x}_c)) = \int p(\mathbf{x}) d\mathbf{x}_n = \int p(\boldsymbol{\phi}(\mathbf{x}_c, \mathbf{x}_n)) d\mathbf{x}_n = \int P(\mathbf{y} \in A | \mathbf{x}, data) d\mathbf{x}_n$$

and R_c is the projection of the feasible region R into the space of controllable factors only. That is, after obtaining $p(\mathbf{x})$ we integrate over the components of $\mathbf{x}$ that we cannot control completely, then we use the resulting density to calculate the probability of being inside the specification region.

If the problem in equation (10) is solved and the value of $p(\mathbf{g}(\mathbf{x}_c))$ at optimality is high, then we have achieved a *robust* process. Such a process will result in products that are highly likely to meet the specifications defined by the region A regardless of the variability introduced by the noise factors $\mathbf{x}_n$. Since the proposed methodology considers implicitly the variance and mean of the responses, there is no need to make any explicit trade-off between mean and variance, as it is commonly done in the Dual Response Problem (Khatree 1996,Lucas 1994, Myers 1992, Myers 1997, Neff and Myers 1999, Vining 1990). Furthermore, the proposed approach allows for correlation between the responses, and it considers the uncertainty of the parameter estimates.

Following, we present a methodology to evaluate the function $p(\mathbf{g}(\mathbf{x}_c))$. Problem (10) can then be solved using this evaluation method and then applying any of the optimization methods discussed in section 3.

1. After performing a suitable experimental design, fit the multivariate model in (1), $\mathbf{Y} = \mathbf{X}\widehat{\mathbf{B}}$.

2. Set the counter $c = 0$.

3. Assuming the distribution of the noise variables is known, simulate a value of $\mathbf{x}_n$. The vector $\mathbf{x}_c$ is given and contains the values of the controllable factors at which we want to evaluate the probability of conformance given the uncertainty in the noise factors, $p(\mathbf{g}(\mathbf{x}_c))$. Substitute the simulated value of $\mathbf{x}_n$ and the given value $\mathbf{x}_c$ in the model and form the vector of regressors $\mathbf{x} = \boldsymbol{\phi}(\mathbf{x}_c, \mathbf{x}_n)$.

3. Since:

$$\mathbf{y}|\mathbf{x}, data \sim \mathbf{T}_q^{\nu}\left(\mathbf{B}'\mathbf{x}, \frac{\nu}{\nu-2}\mathbf{H}^{-1}\right)$$

   where $\mathbf{T}_q$ denotes a q-variate t distribution with ν degrees of freedom, simulate a value of $\mathbf{y}$ using equation (6).

4. If $\mathbf{y} \in A$, make $c \leftarrow c + 1$. Go to step 2 and repeat N times.

6. Return $\widehat{p}(\mathbf{g}(\mathbf{x}_c)) = c/N$ as an estimate of $p(\mathbf{g}(\mathbf{x}_c))$.

5 HPLC Example

The following example was taken from Peterson (2000). It involves the optimization of a high performance liquid chromatography (HPLC) system to detect mixtures of impurities. The performance of the assay was based upon four quantitative response variables, the critical resolution (Rs), total run time, signal-to-noise ratio of the last peak and the tailing factor of the major peak. Three controllable factors affecting the HPLC assay were included: %IPA, temperature and pH. A Box-Behnken design with three center points was used. The data are presented in Table 1. Complete second order response surface models were fitted to all three responses using coded versions of the controllable factors. The coding convention used is the *orthogonal coding* described in (Myers and Montgomery, 1995, pp. 22). The resulting R^2 statistics were higher than 99.7% for all the four models. This indicates that the models fitted the data very well.

The controllable factor %IPA was assumed to be the noise variable, which was further assumed to be normally distributed with mean zero and standard deviation of 0.1.

5.1 Grid-based Method and Meta-Model Approaches

A 11^2 design was carried out in the two remaining regressors, where the levels of the regressors were chosen from:

$$\{-1.0, \quad -0.8, \quad -0.6, \quad -0.4, \quad -0.2, \quad 0.0, \quad 0.2, \quad 0.4, \quad 0.6, \quad 0.8, \quad 1.0\}$$

Table 1: Factor and Response Data for the HPLC Example

Controllable Factors			Responses			
%IPA	Temp.	pH	Rs	Run Time	S/N ratio	Tailing
65	30	0.175	2.14	22	172	0.76
65	50	0.175	1.73	12	311	0.88
65	40	0.050	1.93	16	251	0.80
65	40	0.300	1.95	16	241	0.80
70	40	0.175	2.17	14	278	0.79
70	50	0.050	1.97	11	371	0.86
70	30	0.300	2.38	19	194	0.74
70	50	0.300	1.98	11	360	0.86
70	30	0.050	2.37	18	204	0.74
70	40	0.175	2.20	14	280	0.78
75	40	0.300	2.42	13	314	0.78
75	30	0.175	2.61	17	223	0.73
75	50	0.175	2.14	10	410	0.85
75	40	0.050	2.42	12	324	0.78
70	40	0.175	2.20	14	281	0.79

which required 121 function evaluation. The A region was specified as:

$$A = \left\{ \mathbf{y} = [y_{Rs}\ y_{Time}\ y_{S/N}\ y_{Tail}] : y_{Rs} \geq 1.8,\ y_{Time} \leq 15,\ y_{S/N} \geq 300,\ 0.75 \leq y_{Tail} \leq 0.85 \right\}$$

which is the same region used in Peterson (2000). Five thousand (i.e. N=5000) samples were generated to obtain each value of $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$, which took 0.06 seconds of cpu computing time on MATLAB running on a 800 MHz PC, so the 121 evaluations required 8.5 seconds (this includes some extra time for storage). The highest probability obtained was 0.9660 for $x_1 = 0.4$ and $x_2 = -0.4$ (in the remainder the temperature will be denoted by x_1 and the pH by x_2).

A fourth degree polynomial logistic regression model was fitted to the simulated $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ results using Minitab's Binary Logistic Regression command. Although the Goodness-of-fit statistics rejected the hypothesis of a good fit (all were 0.000), there were 97.7% concordant pairs, 2.0% discordant and 0.3% ties. The "Goodman-Kruskal Gamma" and Somer's D were 0.96 both, while the Kendall's τ was only 0.38.

The fitted model is given by the following equation:

$$\begin{aligned} logit(\widetilde{p}(\mathbf{g}(\mathbf{x}_c))) = \quad & -20.5 + 133.7x_1 - 8.82x_2 + 38.7x_1x_2 - 265.2x_1^2 + 3.38x_2^2 - 53.9x_1^2x_2 \\ & -8.80x_1x_2^2 + 6.12x_1^2x_2^2 + 218.8x_1^3 + 0.112x_2^3 + 24.1x_1^3x_2 - 0.145x_1x_2^3 \\ & -69.14x_1^4 - 0.271x_2^4 \end{aligned} \tag{11}$$

Table 2: Optimal Solutions ($\mathbf{x}_c^*$) using the Meta-Model Approach

$\mathbf{x}_c^*$	$\widetilde{p}(\mathbf{g}(\mathbf{x}_c^*))$	$\bar{p}(\mathbf{g}(\mathbf{x}_c^*))$	$\widehat{\sigma}_{\bar{p}(\mathbf{g}(\mathbf{x}_c^*))}$
$[0.4822, \quad 1]'$	0.9684	0.9622	2.70E-04
$[0.3752, -1]'$	0.9769	0.9681	2.49E-04

Table 3: Probabilities of out-of-bounds for the four responses (Meta-Model Approach)

$\mathbf{x}_c^*$	$P(y_{Rs} \leq 1.8)$	$P(y_{Time} \geq 15)$	$P(y_{S/N} \leq 300)$	$P(y_{Tail.} \leq 0.75)$	$P(y_{Tail.} \geq 0.85)$
$[0.4822, \quad 1]'$	0.0021	0.0166	0.0023	0.0024	0.0228
$[0.3752, -1]'$	0.0019	0.0166	0.0029	0.0031	0.0120

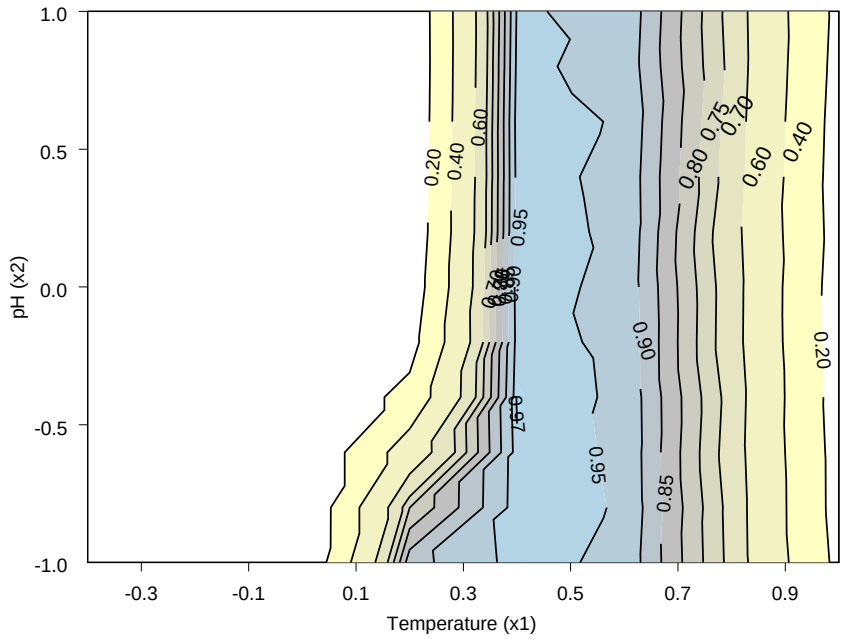
A trust region algorithm was used to maximize the preceding quartic logistic regression. The algorithm was started at 50 different locations chosen randomly inside the experimental region defined by the experimental design used in Table 1. In coded factors this region is $R_c = \{[x_1 \;\; x_2] : -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1\}$.

In all of the 50 optimizations, the algorithm converged only to two different points. These points, denoted by $\mathbf{x}_c^*$ together with the values of $\widetilde{p}(\mathbf{g}(\mathbf{x}_c^*))$ predicted for them by the meta-model in (11) are reported in Table 2. This table also contains estimates of $p(\mathbf{g}(\mathbf{x}_c^*))$ obtained from 500,000 Monte Carlo samples (denoted by $\bar{p}(\mathbf{g}(\mathbf{x}_c))$). Standard errors for $\bar{p}(\mathbf{g}(\mathbf{x}_c^*))$ were calculated using equation (8).

As it can be seen, both solutions given by the meta-model approach are similarly good in terms of the probability value and the final user could choose any of them based on other considerations. However, it is also seen that the meta-model tends to slightly over-estimate the probabilities. Since there are only two controllable factors it is easy to plot the estimates $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ obtained from the Monte Carlo simulations. The plots, presented in Figure 1, show a ridge-type form, indicating that only x_1 (temperature) has an effect on $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$. This is due to the fact that the pH does not interact with the %IPA and, therefore, is not useful for controlling the variance of the response.

Another, possibly more important reason is that the pH does not have any significant effect on the Time and Tailing responses and, as it can be seen in Table 3, these responses did not fall out of the desired region with high frequency during the Monte Carlo simulations. This is equivalent to say that the desired ranges for the responses affected by the pH are easily met and therefore there is almost complete freedom in the value of pH chosen for carrying the HPLC assays.

Finally, it may be seen that choosing the maximum from the grid of points and using the meta-model approach yielded very similar optimal values for x_1 but different for x_2. Clearly this is due to the non-significant effect of this last controllable factor.



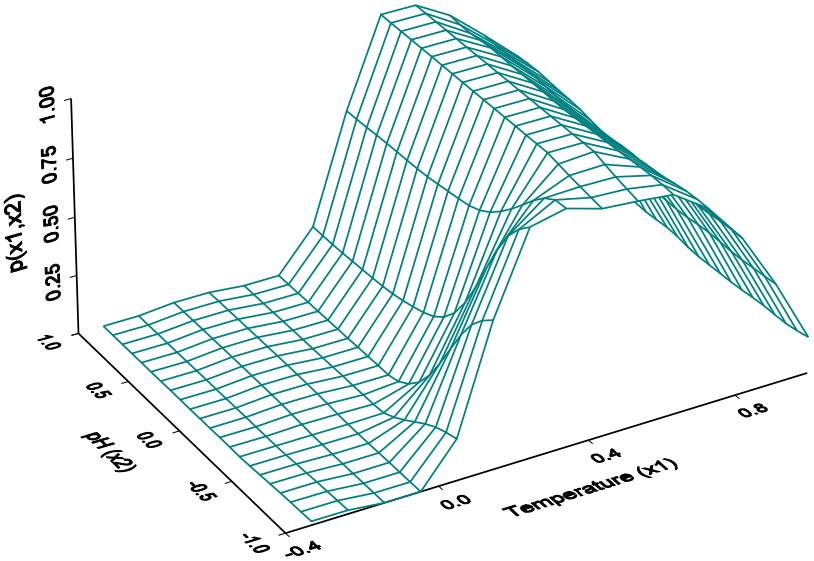


Figure 1: Plots of $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$

5.2 Direct Search Method-Nelder Mead

Matlab's fminsearch routine, which is an implementation of a standard Nelder-Mead algorithm, was used for this optimization. The preliminary optimization to find a good starting point was performed using Matlab's fmincon routine which uses a trust region algorithm. The starting point obtained from solving (9) was $\mathbf{x}_c^o = [0.47\ 0.47]$, for which $\bar{p}(\mathbf{g}(\mathbf{x}_c^o))$ (from 500,000 Monte Carlo samples) was 0.9642. To reduce the variability in the function evaluations feeded into the algorithm, each value of $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ consisted of 20,000 Monte Carlo samples. However, doing this only took ≈ 0.3 seconds.

The main optimization problem in (10) was solved 20 times both by fixing and without fixing the seed of the random number generator. For the fixed-seed cases a different seed was used for each of the 20 trials, but it was maintained constant throughout each complete optimization, that is, until convergence of the Nelder-Mead algorithm was obtained. For the case without fixing the seed, a different stream of random numbers was used for each evaluation of $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$.

The convergence points of the algorithm for both of the aforementioned cases are presented in Figure 2. Tables 5 and 4 contain these convergence points together with the corresponding estimated probabilities, $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$, and the cpu time required for each complete optimization. Comparing these results with Figure 1 we can see that the algorithm stopped in points were the probability is high, and that the difference in the optimal values of $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ is small. It is also observed that for the varying seed case the algorithm moved very little from the starting point. However, for the fixed seed case, the algorithm significantly changed x_2 from the original value of 0.47 to somewhere in the $(-0.56,\ -0.8)$ range, regardless of the seed. Notice that this resulted in slightly higher probabilities.

Clearly, the initial solution is very good, specially since its sets the highly critical x_1 to a value close to 0.4. This leaves very little room for the Nelder-Mead to improve. Therefore, to test the efficacy of the Nelder-Mead algorithm in case the initial point is not so good, the optimization was started at randomly selected points inside the R_c region (for which the seed was fixed). These results are presented in Table 6 and it can be observed that either the algorithm converged to a very good solution or it was exactly zero. The problem is that the function being optimized is too flat in the vicinities of a solution for which $\widehat{p}(\mathbf{g}(\mathbf{x}_c)) = 0$ and therefore the algorithm is unable to find an improving direction.

Regarding the time requirements of both methods, it is clear that the time spent in obtaining the data for building the meta-model was less than the direct search approach. However, the time required to find a model with good fit cannot be assessed quantitatively. In addition, we notice that for the direct search with fixed seeds the number of function evaluations was cut by half in most cases, while the resulting probabilities were similar.

Table 4: Optimal solutions for example 1 using the Nelder-Mead Algorithm with a different seed for the random number generator in each trial. In each trial, all $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ estimates were obtained with the same stream of random numbers. The initial point for the Nelder-Mead algorithm was the solution of problem (9).

	Optimal Solution			cpu time	
Trial	x_1	x_2	$\widehat{p}(\mathbf{g}(\mathbf{x}^*{}_c))$	(sec.)	Evaluations
1	0.4355	-0.5607	0.9712	15.5	51
2	0.4382	-0.8063	0.9711	18.5	61
3	0.4311	-0.5953	0.9706	15.5	51
4	0.4109	-0.7000	0.9713	18.0	59
5	0.4201	-0.5126	0.9689	14.2	47
6	0.4302	-0.7563	0.9702	19.1	63
7	0.4083	-0.7917	0.9697	18.5	61
8	0.4298	-0.5774	0.9702	16.2	54
9	0.4270	-0.7128	0.9691	20.3	67
10	0.4123	-0.5464	0.9707	13.9	46
11	0.4090	-0.7589	0.9721	19.3	64
12	0.4389	-0.5477	0.9709	16.3	54
13	0.4204	-0.7787	0.9702	18.7	62
14	0.4473	-0.5343	0.9694	16.1	53
15	0.4019	-0.7938	0.9694	20.4	67
16	0.4348	-0.5347	0.9694	13.4	44
17	0.4285	-0.5760	0.9702	16.6	55
18	0.4201	-0.5273	0.9698	13.6	45
19	0.4296	-0.7032	0.9702	19.0	63
20	0.4189	-0.5729	0.9691	15.8	52

Table 5: Optimal solutions for example 1 using the Nelder-Mead Algorithm without fixing the seed for the random number generator in each trial. In all the trials, the initial point for the Nelder-Mead algorithm was the solution of problem (9).

	Optimal Solution			cpu time	
Trial	x_1	x_2	$\widehat{p}(\mathbf{g}(\mathbf{x}^*{}_c))$	(sec.)	Evaluations
1	0.4700	0.4966	0.9678	15.7	52
2	0.4913	0.4708	0.9666	12.2	40
3	0.4637	0.4840	0.9662	8.1	27
4	0.4647	0.4905	0.9677	25.6	85
5	0.4694	0.4934	0.9677	15.5	51
6	0.4583	0.4818	0.9675	108.7	362
7	0.4677	0.4682	0.9674	21.6	71
8	0.4796	0.4598	0.9666	9.4	31
9	0.4710	0.4950	0.9673	39.7	132
10	0.4705	0.4715	0.9674	21.3	71
11	0.4733	0.4848	0.9681	21.5	71
12	0.4882	0.4829	0.9681	109.2	363
13	0.4752	0.4873	0.9680	64.4	214
14	0.4704	0.4614	0.9676	29.1	96
15	0.4775	0.4821	0.9673	11.1	37
16	0.4884	0.4811	0.9659	7.9	26
17	0.4740	0.4912	0.9680	81.2	270
18	0.4824	0.4838	0.9681	109.7	365
19	0.4917	0.4957	0.9671	109.1	363
20	0.4876	0.4953	0.9679	108.5	361

Table 6: Optimal solutions for example 1 using the Nelder-Mead Algorithm with a different seed for the random number generator in each trial. In each trial, all $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ estimates were obtained with the same stream of random numbers. The initial point for the Nelder-Mead algorithm was randomly selected within the region R_c.

	Optimal Solution			cpu time
Trial	x_1	x_2	$\widehat{p}(\mathbf{g}(\mathbf{x}^*{}_c))$	(sec.)
1	-0.1369	-0.4789	0	14.7
2	0.4760	0.4439	0.9654	51.0
3	-0.3366	-0.9845	0	13.6
4	0.4856	0.3597	0.9627	18.9
5	-0.9473	-0.2891	0	13.4
6	-0.4027	-0.6376	0	12.6
7	0.4669	0.3056	0.964	17.7
8	-0.1918	-0.3961	0	10.3
9	-0.3817	-0.9935	0	10.9
10	0.4709	0.3858	0.9656	15.7
11	-0.6747	0.1041	0	11.5
12	0.4482	-0.1723	0.9674	18.8
13	0.4125	-0.7858	0.9691	23.5
14	0.4775	0.5173	0.9617	28.8
15	0.4765	0.6786	0.9649	19.7
16	-0.6467	0.2335	0	12.9
17	-0.2710	0.5466	0	12.9
18	-0.3632	0.1920	0	11.0
19	-0.9328	0.7140	0	11.5
20	0.4764	0.5058	0.9650	26.1

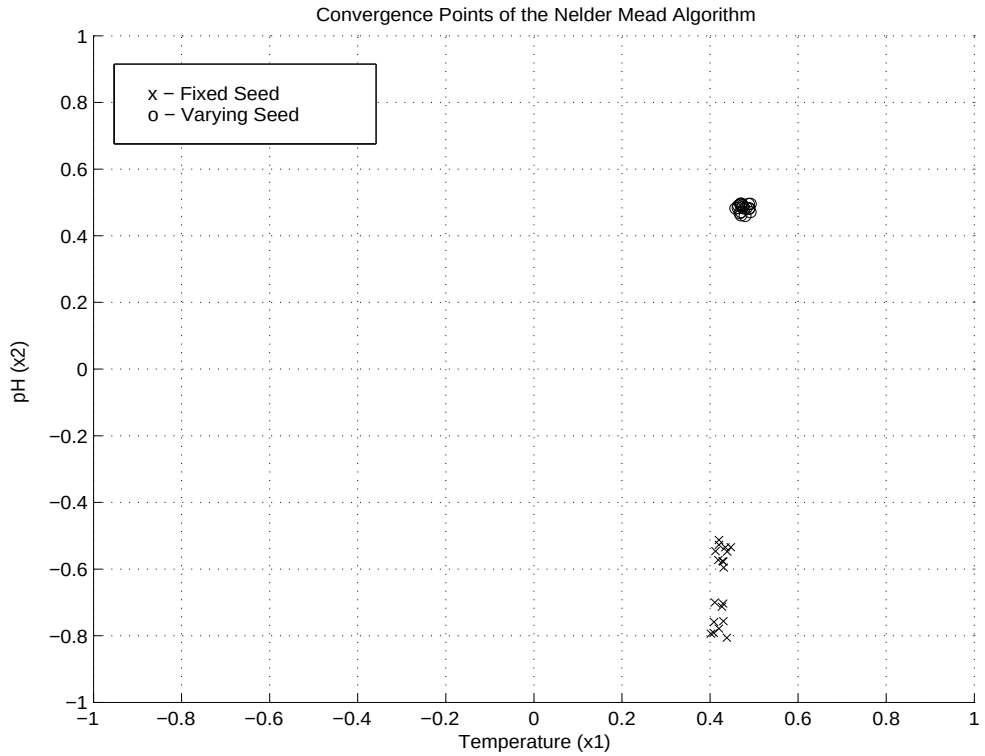


Figure 2: Convergence points of the Nelder-Mead Algorithm. Example 1.

6 Chemical Process Example

This example was taken from Wold et al. (1989) and consists of an experiment to reduce the amount of by-products from a chemical process. There are 5 controllable factors, 5 responses and the experimental design used is a 2^{5-1} fractional factorial with two center points. The data are given in Table 7.

First order models with two-factor interactions were fitted to all the responses. It was found that x_3 was not significant in any of the responses and was dropped from all the models. This has the advantage of producing a full 2^4 factorial design in the remaining factors. In addition, the model for y_1 had a poor fit ($R^2 = 64.6\%$) and therefore was also dropped from the analysis. The R^2 statistics for the remaining responses were: 87.0% for y_2, 96.3% for y_3, 93.3% for y_4 and 85.7% for y_5. Notice that the models for this example present a poorer fit than for the first example. The remaining data set consists of four controllable factors and four responses.

To perform the proposed methodology, x_1 was assumed to be a noise variable and to be normally distributed with mean zero and standard deviation equal to 0.1. In addition, the region of acceptance, A, was defined by

$$A = \{\mathbf{y} = [y_2\ y_3\ y_4\ y_5] : y_2 \geq 91.0,\ y_3 \leq 11.5,\ y_4 \leq 6.5, \leq y_5 \leq 5.5\}$$

Table 7: Experimental Results for the Chemical Process

x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4	y_5
-1	1	-1	-1	-1	80.0	93.7	5.1	1.2	2.6
-1	1	-1	1	1	80.0	88.7	10.9	0.4	4.0
1	-1	-1	1	1	91.0	90.8	9.0	0.2	1.9
1	-1	-1	-1	-1	86.0	94.3	3.5	2.2	1.2
0	0	0	0	0	75.0	92.7	7.1	0.2	2.5
1	1	1	-1	-1	89.0	95.0	4.4	0.6	1.4
-1	1	1	1	-1	84.0	91.7	8.3	0.0	2.4
-1	-1	1	-1	-1	80.0	82.8	2.3	14.9	0.6
-1	-1	1	1	1	83.0	90.0	4.1	5.9	0.7
-1	1	1	-1	1	84.0	94.6	5.4	0.0	1.5
1	-1	1	-1	1	89.0	96.2	3.8	0.0	1.6
1	-1	1	1	-1	92.0	94.5	5.5	0.0	1.5
0	0	0	0	0	96.0	94.1	5.9	0.0	2.8
1	1	1	1	1	88.0	86.9	13.9	0.0	7.9
-1	-1	-1	1	-1	89.0	81.2	4.8	14.0	2.6
-1	-1	-1	-1	1	81.0	87.4	4.1	8.5	0.5
1	1	-1	-1	1	100.0	92.1	7.9	0.0	5.1
1	1	-1	1	-1	90.0	89.3	10.7	0.0	9.2

The region X, where the models are assumed valid, was taken as $-1.0 \leq x_i \leq 1.0$, $\forall i = 2, 4, 5$.

6.1 Grid Approach

A 11^3 design was carried out in the three remaining regressors, the levels of the regressors were chosen from:

$$\{-1.0, \quad -0.8, \quad -0.6, \quad -0.4, \quad -0.2, \quad 0.0, \quad 0.2, \quad 0.4, \quad 0.6, \quad 0.8, \quad 1.0\}$$

This requires 1331 evaluations. Since no meta-model was fit, 20000 simulations were used to obtain each of the $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ estimates. The argument is that whenever a meta-model is used it serves as a smoothing surface and therefore more uncertainty could be allowed in the individual determinations. The total cpu time for obtaining the data was ≈ 451 seconds.

The current problem has three controllable factors, and therefore, it is not easy to present a graphical representation of the $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ values. The plot in Figure 3 is a 3-D scatter plot where the diameter of the circles and the intensity in the gray scale is proportional to $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ (darker color implies higher value). After a careful inspection, it may be noticed that the probabilities are high around $\mathbf{x}_c = [1 \ -1 \ -1]$ and around $\mathbf{x}_c = [-1 \ 1 \ -1]$, where the value of x_2 seems to be the less relevant. It is also observed that the probabilities are low around $\mathbf{x}_c = [1 \ 1 \ 1]$ and $\mathbf{x}_c = [-1 \ -1 \ -1]$. In addition the function is clearly not convex.

The highest probability obtained from the grid values is $\widehat{p}(\mathbf{g}(\mathbf{x}_c^*)) = 0.8133$ for $\mathbf{x}_c^* = [1 \ -1 \ -1]$. Performing the Monte Carlo with 500,000 simulations at this location we get $p(\mathbf{g}(\mathbf{x}_c)) = 0.8087$.

6.2 Direct Search Method (Nelder-Mead) and Sequential Quadratic Programming

The initial solution, obtained by solving problem (9), was $\mathbf{x}_c^o = [0.0324 \ \ 0.0157 \ \ -0.0266]$, with $\bar{p}(\mathbf{g}(\mathbf{x}_c^o)) = 0.4517$ (from 500,000 Monte Carlo samples). It can be seen that in this case there is considerable room for improvement over the initial solution. However, the initial solution obtained is much better than the worst solution observed from the grid approach, $\widehat{p}(\mathbf{g}(\mathbf{x}_c)) = 0.0480$ for $\mathbf{x}_c = [1 \ 1 \ 1]$.

As in example 1, the optimization was performed 20 times and 20,000 Monte Carlo samples were generated to obtain each value of $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$. However, in this example the seed of the random number generator was always fixed for each trial, but as before, a different seed was used in different trials. The convergence points obtained are reported in Table 8. It can be seen that in the majority of the cases the probability obtained at optimality is slightly lower to the one obtained using the grid. The problem may be related to the fact that more than one of the bounds is active at the optimal point and the penalty functions induce an early termination of the algorithm (Bazaraa et al., 1993, p.369). This is worsened by the fact that the effect of x_2 is very small, as pointed out earlier.

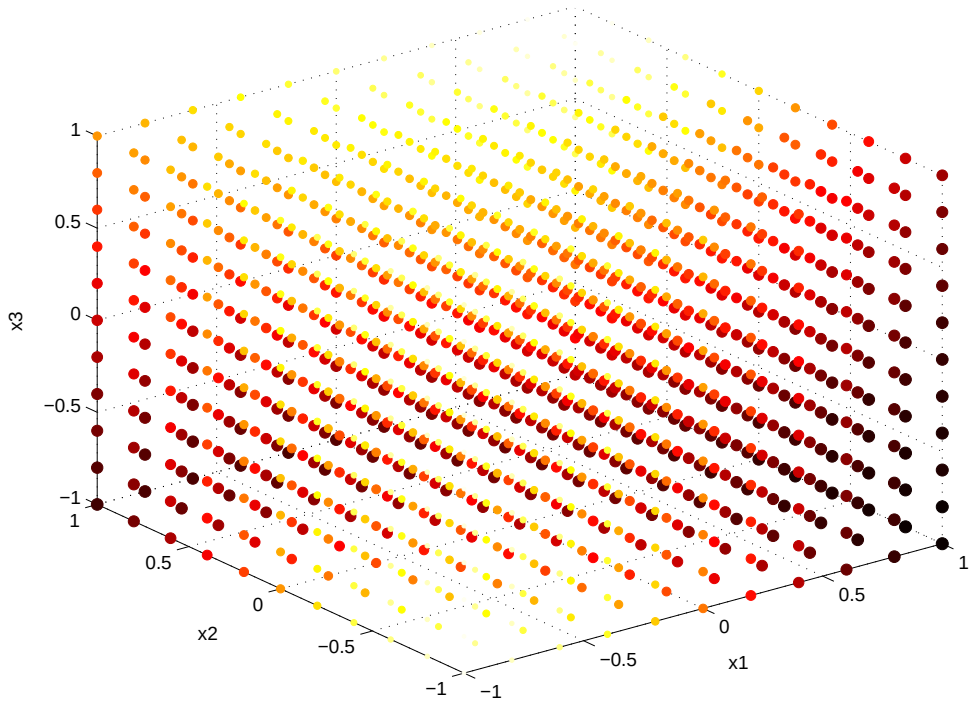


Figure 3: 3-D Scatter plot for the Grid Approach

To avoid the use of penalty functions, Matlab's fmincon routine was used which is designed for constrained optimization. This routine uses a SQP algorithm in the absence of gradient information. The results, included in Table 9, indicate that this algorithm converged to the optimal solution in all of the 20 trials. Furthermore, compared to the grid approach, it only required an average of 31.55 function evaluations, 42 times lower than the grid approach and it required on average 44 times less time. Finally, all the optimal solutions were equal because all of the controllable factors are at one of their bounds and the bias introduced by holding the seed constant is small enough not to impact the quality of the estimates. The range of the $\widehat{p}(\mathbf{g}(\mathbf{x}^*{}_c))$ is just 0.0102 or 1%.

Table 8: Optimal solutions for example 2 using the Nelder-Mead Algorithm with a different seed for the random number generator in each trial. In each trial, all $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ estimates were obtained with the same stream of random numbers. The initial point for the Nelder-Mead algorithm was the solution of problem (9).

	Optimal Solution				cpu time	
Trial	x_2	x_4	x_5	$\widehat{p}(\mathbf{g}(\mathbf{x}^*{}_c))$	(sec.)	Evaluations
1	0.9999	-0.2881	-0.9999	0.7793	34.6	115
2	0.9976	-0.2876	-0.9999	0.7823	40.7	130
3	0.9994	-0.2310	-0.9998	0.7777	45.0	145
4	0.9999	-0.2871	-0.9999	0.7846	32.2	107
5	0.9997	-0.3276	-0.9985	0.7844	55.5	171
6	1.0000	-0.4930	-0.6857	0.7626	36.8	108
7	1.0000	-0.4942	-0.4060	0.7446	30.3	89
8	0.9999	-0.8030	-0.9989	0.8056	74.9	219
9	0.9840	-0.5858	-0.9998	0.7940	55.0	162
10	0.9982	-0.6078	-0.7053	0.7818	38.1	111
11	1.0000	-0.4088	-0.9878	0.7870	33.7	99
12	0.4542	-0.2396	-0.9999	0.6902	34.3	101
13	0.7336	-0.2067	-1.0000	0.7416	38.4	112
14	1.0000	-0.4852	-0.7220	0.7712	41.1	121
15	0.9998	-0.2871	-1.0000	0.7785	41.2	120
16	1.0000	-0.0802	-0.4483	0.6896	39.2	115
17	1.0000	-0.5892	-0.6726	0.7705	37.4	109
18	0.9999	-0.7299	-0.9995	0.7997	39.5	116
19	0.8947	-0.6364	-0.9993	0.7805	67.9	199
20	0.9999	-0.1622	-0.4913	0.7030	35.8	105

Table 9: Optimal solutions for example 2 using the SQP Algorithm with a different seed for the random number generator in each trial. In each trial, all $\widehat{p}(\mathbf{g}(\mathbf{x}_c))$ estimates were obtained with the same stream of random numbers. The initial point for the SQP algorithm was the solution of problem (9).

	Optimal Solution				cpu time	
Trial	x_2	x_4	x_5	$\widehat{p}(\mathbf{g}(\mathbf{x}^*{}_c))$	(sec.)	Evaluations
1	1.0000	-1.0000	-1.0000	0.8046	9.2	29
2	1.0000	-1.0000	-1.0000	0.8086	6.8	21
3	1.0000	-1.0000	-1.0000	0.8045	10.7	33
4	1.0000	-1.0000	-1.0000	0.8122	10.7	33
5	1.0000	-1.0000	-1.0000	0.8085	9.4	29
6	1.0000	-1.0000	-1.0000	0.8082	12.2	37
7	1.0000	-1.0000	-1.0000	0.8070	9.4	29
8	1.0000	-1.0000	-1.0000	0.8076	12.0	37
9	1.0000	-1.0000	-1.0000	0.8087	9.4	29
10	1.0000	-1.0000	-1.0000	0.8118	8.1	25
11	1.0000	-1.0000	-1.0000	0.8124	9.4	29
12	1.0000	-1.0000	-1.0000	0.8134	12.0	37
13	1.0000	-1.0000	-1.0000	0.8077	12.0	37
14	1.0000	-1.0000	-1.0000	0.8068	10.9	33
15	1.0000	-1.0000	-1.0000	0.8056	14.2	44
16	1.0000	-1.0000	-1.0000	0.8091	8.2	25
17	1.0000	-1.0000	-1.0000	0.8076	12.0	37
18	1.0000	-1.0000	-1.0000	0.8147	10.7	33
19	1.0000	-1.0000	-1.0000	0.8100	10.7	33
20	1.0000	-1.0000	-1.0000	0.8101	6.9	21

7 Conclusions

A procedure proposed by Peterson (2001) for multiple response optimization using a predictive density approach was extended to include noise variables. The extra integration over the probability distribution of the noise variables is done by Monte Carlo simulation together with the simulation of the response vectors. A computer code written with currently available software (Matlab 6.1) is able to perform 50,000 of such simulations in ≈ 0.7 seconds on a Pentium III at 800 MHz.

An optimization scheme was also proposed that finds an initial solution by minimizing the scaled prediction variance of the model subject to the expected response being inside the desirable region. This initial solution provided a location where the response functions have slopes steep enough to ease the subsequent optimization problem.

Two optimization algorithms were used. A standard Nelder-Mead algorithm and a Sequential Quadratic Programming (SQP) algorithm. Constraints where handled using penalty functions in the Nelder-Mead implementation. The Nelder-Mead method was used both by fixing and without fixing the seed for the random number generator. Fixing the seed eliminates the variability between the Monte Carlo simulations, which substantially enhances the performance of the optimization algorithms but produces results that depend on the seed used. It was observed that fixing the seed produced better results and that the large number of replications that the fast computation allowed produced an insignificant bias.

The SQP algorithm was used only with a fixed seed and converged to the optimal solution in all the cases observed. SQP significantly outperformed Nelder-Mead in the case when the optimal solution was constrained.

8 References

Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (1993). *Nonlinear Programming, Theory and Algorithms*. 2^{nd} Ed. Wiley. New York, NY.

Box, G.E.P. and Jones, S. (1990). "Designing Products That Are Robust To The Environment. *Report Series in Quality and Productivity*. CPQI, University of Wisconsin, Number 56.

Chiao, C. and Hamada, M. (2001). "Analyzing Experiments with Correlated Multiple Responses". *Journal of Quality Technology*, 33, pp. 451-465.

Derringer, G. and Suich, R. (1980). "Simultaneous Optimization of Several Response Variables". *Journal of Quality Technology*, 12, pp. 214-219.

Harrington, E.C. (1965). "The Desirability Function". *Industrial Quality Control*, 21, pp. 494-498.

Johnson, M.E. (1987). *Multivariate Statisitcal Simulation*. Wiley. New York, NY.

Khatree, R. (1996). "Robust Parameter Design: A Response Surface Approach". *Journal of Quality Technology*, 28, pp. 187-198.

Lucas, J.M. (1994). "How to Achieve a Robust Process Using Response Surface Methodology". *Journal of Quality Technology*, 26, pp. 248-260.

Myers, R.H., Khuri, A.I. and Vining G. (1992). "Response Surface Alternatives to the Taguchi Robust Parameter Design". *The American Statistician*, 46, pp. 131-139.

Myers, R.H. and Montgomery, D.C. (1995). *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. John Wiley and Sons, New York, NY.

Myers, R.H., Kim, Y. and Griffiths, K.L. (1997). "Response Surface Methods and the use of Noise Variables". *Journal of Quality Technology*, 29, pp. 429-440.

Neff, A.R. and Myers, R.H.(1999). "Recent Developments in Response Surface Mthodology and its Applications in Industry". In: *Statistical Process Monitoring and Optimization*, Sung H. Park and G.G. Vining, Eds., New York: Marcel Dekker, pp. 457-481.

Nelder J.A. and Mead, R. (1964). "A Simplex Method for Function Minimization", *Computer Journal*, 7, pp. 308-313.

Peterson, J.J. (2000). "A Probability-based Desirability Function for Multiresponse Optimization". Proceedings of the Section on Quality and Productivity, Annual Meeting of the American

Statistical Association, Indianapolis, IN.

Press, S.J. (1982). *Applied Multivariate Analysis: Using Bayesian and Frequentist Methods of Analysis*. R. E. Krieger Pub. Co. Malabar, FL.

Press, S.J. (1989). *Bayesian Statistics: Principles, Models and Applications*. Wiley. New York, NY.

Rubinstein, R.V. (1981). *Simulation and the Monte Carlo Method*. Wiley. New York, NY.

Taguchi, G. (1986). *Introduction to Quality Engineering*. Quality Resources, White Plains, NJ.

Taguchi, G. (1987). *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Cost*. Quality Resources, White Plains, NJ.

Taguchi, G. and Wu, Y. (1985). *Introduction to Off-Line Quality Control*. Central Japan Quality Control Association (available from American Supplier Institute, Dearborn, MI).

Vining, G.G. and Myers, R.H. (1990). "Combining Taguchi and response Surface philosophies: A dual response aproach".*Journal of Quality Technology*, 22, pp. 38-45.

Wold, S., Carlson, R. and Skagerberg, B (1989). "Statistical Optimization as a Means to Reduce Risks in Industrial Processes" *The Environmental Professional*, 11, pp. 127-131.

Wu, C.F.F and Hamada, M. (2000). *Experiments, Planning, Analysis and Parameter Design Optimization*. John Wiley and Sons, New York, NY.