

Space-Filling Designs for Robustness Experiments

V. Roshan Joseph

H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology, Atlanta, GA 30332

Li Gu

Amazon.com, Inc.,
Seattle, WA 98121

Shan Ba and William R. Myers

Quantitative Sciences
The Procter & Gamble Company, Mason, OH 45040

Abstract

To identify the robust settings of the control factors, it is very important to understand how they interact with the noise factors. In this article, we propose space-filling designs for computer experiments that are more capable of accurately estimating the control-by-noise interactions. Moreover, the existing space-filling designs focus on uniformly distributing the points in the design space, which are not suitable for noise factors because they usually follow non-uniform distributions such as normal distribution. This would suggest placing more points in the regions with high probability mass. However, noise factors also tend to have a smooth relationship with the response and therefore, placing more points towards the tails of the distribution is also useful for accurately estimating the relationship. These two opposing effects make the experimental design methodology a challenging problem. We propose optimal and computationally efficient solutions to this problem and demonstrate their advantages using simulated examples and a real industry example involving a manufacturing packing line.

KEY WORDS: Computer experiments, Experimental design, Gaussian process, Optimal designs, Quality improvement, Robust parameter design.

1 INTRODUCTION

Robust parameter design is a cost-efficient technique for quality improvement. Originally proposed by Taguchi (1987), the technique has been widely adopted in industries for system (product or process) optimization. The core idea is to first divide the factors in the system into two groups: control factors and noise factors. Control factors are those factors in the system than can be cost-effectively controlled. On the other hand, noise factors are those factors which are either impossible or too expensive to control. For example, in product design of a razor for shaving, blade thickness, gap between the blades, angle of the blades, etc. are control factors, whereas consumer attributes like the skin type, hair length and density, and product usage attributes like the pressure applied on the skin, handle angle, etc. are noise factors. Since the noise factors are uncontrollable, they introduce variability in the performance of the product. Robust parameter design is a technique to find a setting of the control factors (also known as parameter design) that will make the system robust or insensitive to the noise factors. Thus, under a robust parameter design, the output becomes less affected by the noise variability even when the noise factors are left uncontrolled. This is why the approach using robust parameter design is less costly than the other quality improvement techniques, which try to directly control the noise factors in the system.

The key to a successful robust parameter design is in identifying important control-by-noise interactions of the system. Only when such interactions exist we can use the control factors to reduce the sensitivity of the noise factors. These interactions are usually unknown in practice and their existence need to be investigated through experimentation. Thus designing good experiments is a crucial step in robustness studies. Many efficient experimental design techniques are proposed in the literature such as cross arrays (Taguchi, 1987) and single arrays (Welch et al., 1990; Shoemaker et al., 1991; Wu and Zhu, 2003; Kang and Joseph, 2009). A thorough discussion of these techniques can be found in the books by Wu and Hamada (2009) or Myers et al. (2016).

The aforementioned experimental design techniques are mainly proposed for physical experimentation except for the work of Welch et al. (1990). Recently computer experiments have become very common in industry. That is, if a computer model is available that can

simulate the physical system, then the experiments can be performed in computers instead of the physical system. It is becoming very common for industry to develop a computer model for product design like simulating the performance of a razor. One particular example from Procter & Gamble involving the development of a computer model that simulates a critical transformation of a packing line, that involves both control and noise factors, will be discussed in more detail later in the paper. Computer experiments can bring in tremendous cost savings because direct experimentation with the real physical system is always more expensive than investing on some computer time. However, there are several aspects of computer experiments that necessitate the use of a different experimental design technique or philosophy compared to those of physical experiments (Sacks, Welch, Mitchell and Wynn, 1989). Since most computer models are deterministic in nature, randomization and replications are not needed. Fractional factorial and orthogonal array-based design techniques that are prevalent in physical experiments lead to replications when projected onto subspace of factors and thus are unsuitable for computer experiments. Split-plot designs that are considered to be useful in robustness studies (Bingham and Sitter, 2003) become unnecessary as run orders and restrictions on randomization will not affect the computer model outputs. This led to the development of space-filling designs in computer experiments.

The existing work on robust parameter design using space-filling designs do not make any distinction between control and noise factors. A distinction is made only at the analysis stage (Welch et al., 1990; Chen et al., 2006; Apley et al., 2006; Bates et al., 2006; Tan, 2015). Sequential designs that directly attempt to find robust settings of control factors using expected improvement-type algorithms are proposed in the literature (Williams et al., 2000; Lehman et al., 2004), but we are not aware of any work on space-filling designs. It is important to develop space-filling designs that distinguish control and noise factors because their distributional properties are entirely different. Control factors are assumed to follow a uniform distribution, whereas noise factors typically follow non-uniform distributions such as normal distribution. Their nature of randomness is also different. Noise factors are intrinsically random and can vary over time and space. On the other hand, control factors remain fixed once their levels are chosen. A uniform distribution is imposed on the control factors only to represent our indifference on the choice of level given the range of possible values for

each control factor. Thus, unlike the control factors, most of the “action” in the noise factor space takes place in the regions of high probability mass. Therefore, space-filling designs that uniformly spread out points in the experimental region are not adequate for robust parameter design experiments. Moreover, the existing space-filling designs are not designed for precise estimation of control-by-noise interactions. In this article we propose a new version of space-filling design that is capable of estimating the control-by-noise interactions more precisely and puts more points in regions that matters the most.

The article is organized as follows. In Section 2, we propose a model-based optimal experimental design for robustness studies. Because of certain computational and practical difficulties associated with this approach, in Section 3, we propose a modified space-filling design as an alternative. In Section 4, we investigate the optimal choice of noise levels for the experiment. Extension of designs to deal with internal noise factors is proposed in Section 5. The proposed methodology is applied in a simulated example and the packing line computer experiment from Procter & Gamble in Section 6. We conclude with some remarks in Section 7.

2 MODEL-BASED OPTIMAL DESIGNS

Let $\mathbf{x} = (x_1, \dots, x_p)'$ be the set of control factors and $\mathbf{z} = (z_1, \dots, z_q)'$ the set of (external) noise factors. The case of internal noise factors and the differences between the two types will be discussed later. We assume that $\mathbf{x} \in \mathcal{X} = [0, 1]^p$ and $\mathbf{z} \in \mathcal{Z}$, the support of the distribution of \mathbf{z} which could be \mathbb{R}^q . The response y is a deterministic function of both control and noise factors given by $y = g(\mathbf{x}, \mathbf{z})$. Depending on the type of characteristic such as smaller-the-better, larger-the-better, or nominal-the-best, we can impose a quality loss function on y . Let $L(y)$ be such a loss function. Then, the objective of robust parameter design is to find the setting of control factors that minimizes the expected loss, where the expectation is taken with respect to the distribution of noise factors. Let $f(\mathbf{z})$ denote the probability density function of \mathbf{z} . Then, the robust parameter design can be obtained by

$$\min_{\mathbf{x} \in \mathcal{X}} \int_{\mathcal{Z}} L\{g(\mathbf{x}, \mathbf{z})\} f(\mathbf{z}) d\mathbf{z}. \quad (1)$$

Since the function $g(\cdot, \cdot)$ is available only as a computer code, an experiment will be conducted to estimate it. Let $\mathbf{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_p, \mathbf{z}_1, \dots, \mathbf{z}_q\}$ be the experimental design with n runs, where $\mathbf{x}_j = (x_{1j}, \dots, x_{nj})'$ and $\mathbf{z}_k = (z_{1k}, \dots, z_{nk})'$ denote the settings of the j th and k th control and noise factors, respectively. Let y_i be the i th output from the computer model, $i = 1, \dots, n$.

A Gaussian process model or kriging (Santner et al., 2003) is commonly used for estimating $g(\cdot, \cdot)$. So assume

$$g(\mathbf{x}, \mathbf{z}) \sim GP(\mu, \tau^2 R(\cdot, \cdot)), \quad (2)$$

where μ and τ^2 are the unknown mean and variance parameters, and $R(\cdot, \cdot)$ is the correlation function. A commonly used correlation function is the Gaussian correlation function given by

$$R(\mathbf{x}_i - \mathbf{x}_j, \mathbf{z}_i - \mathbf{z}_j) = \exp\left\{-\sum_{l=1}^p \theta_l^x (x_{il} - x_{jl})^2 - \sum_{l=1}^q \theta_l^z (z_{il} - z_{jl})^2\right\},$$

where $\boldsymbol{\theta}^x = (\theta_1^x, \dots, \theta_p^x)'$ and $\boldsymbol{\theta}^z = (\theta_1^z, \dots, \theta_q^z)'$ are the unknown correlation parameters of the control and noise factors, respectively. Let $\boldsymbol{\theta}$ be a $p + q$ column vector containing $\boldsymbol{\theta}^x$ and $\boldsymbol{\theta}^z$. We will use this correlation function throughout this article, but other correlation functions are also allowed as long as they can produce smooth realizations of the response in the noise factor space. The smoothness assumption with respect to noise factors is critical for our methodology and we will exploit it for developing the experimental designs. The Gaussian process can be viewed as a prior on the unknown function and therefore, we can obtain its posterior distribution using Bayes theorem (Santner et al., 2003):

$$g(\mathbf{x}, \mathbf{z}) | \mathbf{y} \sim N\left(\hat{g}(\mathbf{x}, \mathbf{z}), \tau^2 \text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})\right), \quad (3)$$

where $\hat{g}(\mathbf{x}, \mathbf{z}) = \mu + \mathbf{r}(\mathbf{x}, \mathbf{z})' \mathbf{R}^{-1} (\mathbf{y} - \mu \mathbf{1})$ and

$$\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta}) = 1 - \mathbf{r}(\mathbf{x}, \mathbf{z})' \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}, \mathbf{z}),$$

where $\mathbf{y} = (y_1, \dots, y_n)'$, $\mathbf{r}(\mathbf{x}, \mathbf{z})$ is an $n \times 1$ vector with i th element $R(\mathbf{x} - \mathbf{x}_i, \mathbf{z} - \mathbf{z}_i)$, \mathbf{R} is an $n \times n$ matrix with ij th element $R(\mathbf{x}_i - \mathbf{x}_j, \mathbf{z}_i - \mathbf{z}_j)$, and $\mathbf{1}$ is a vector of 1's having length

n. For simplicity, we chose to ignore the extra variability due to the estimation of μ and $\boldsymbol{\theta}$. See Müller et al. (2012) and Müller et al. (2014) for a discussion on the effect of this extra variability on experimental design.

The posterior mean $\hat{g}(\mathbf{x}, \mathbf{z})$ can be used as an estimate of the response function from the experiment (also known as metamodel, surrogate model, or emulator). Then, the optimization in (1) can be simplified as

$$\min_{\mathbf{x} \in \mathcal{X}} \int_{\mathcal{Z}} L\{\hat{g}(\mathbf{x}, \mathbf{z})\} f(\mathbf{z}) d\mathbf{z}. \quad (4)$$

It is also possible to incorporate the uncertainties in the estimation of $g(\cdot, \cdot)$ in the optimization as in Apley and Kim (2011) and Tan and Wu (2012), but it will not be considered here for the sake of simplicity. The problem we are trying to solve is how to design the experiment \mathbf{D} so that we can accurately estimate the solution to the optimization problem in (4).

Clearly, the optimization in (4) will give the true robust setting if $\hat{g}(\mathbf{x}, \mathbf{z})$ is the true response surface, that is, if $\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta}) = 0$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$. Thus we should design the experiment so that $\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})$ is as small as possible. Furthermore, a careful examination of (4) reveals an important insight on the experimental design problem. We need an accurate $g(\cdot, \cdot)$ only in the regions of \mathbf{z} where $f(\mathbf{z})$ is large. In other words, if $f(\mathbf{z})$ is small in some regions, then the inaccuracies in the estimation of $g(\cdot, \cdot)$ in those regions will not affect the robust settings. This makes the experimental design problem for robustness different from that of a usual computer experiment. In fact, it makes sense to focus on the estimated solution to the optimization problem in (4) as proposed in Ginsburg and Ben-Gal (2006) rather than the estimation of $g(\cdot, \cdot)$. However, their approach works only for linear models fitted to physical experimental data. In contrast, the models considered in computer experiments are highly nonlinear and thus, finding an explicit solution to (4) is not feasible. So in this work we will focus on the estimation of $g(\cdot, \cdot)$. Although this approach may not look ideal for the robustness objective, it does have certain advantages. The loss functions are many times loosely defined and one may want to investigate solutions to different possible choices of loss function (Joseph, 2004). Moreover, in real problems, there can be multiple quality characteristics and thus one may need to be satisfied with a compromise solution,

which can be different from the optimal solution in (4). Thus an accurate $g(\cdot, \cdot)$ in the region of interest can be more beneficial than an accurate solution to (4) obtained for a specific choice of loss function and quality characteristic.

Thus, our aim is to find \mathbf{D} such that $\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})$ is small. However, since $\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})$ is a function of \mathbf{x} and \mathbf{z} , it is not possible to find such a design over the entire experimental region. Instead, a feasible approach is to minimize the average of $\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})$, that is

$$\min_{\mathbf{D}} \int_{\mathcal{X}} \int_{\mathcal{Z}} \text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta}) f(\mathbf{z}) d\mathbf{z} d\mathbf{x}. \quad (5)$$

This design criterion is the same as the integrated mean squared error criterion in the literature (Sacks, Welch, Mitchell and Wynn, 1989; Santner et al., 2003) except that we use the density of \mathbf{z} as a weight function. This is quite a natural modification of the existing criterion and agrees with our intuition that we should give more weights for regions where $f(\mathbf{z})$ is large. Surprisingly, we found that the solution to (5) places points in extremely low probability regions which are not very useful for finding the robust setting. This problem can be alleviated if we use the root-mean squared prediction error, which directly corresponds to confidence intervals of the prediction. Thus, consider a modified criterion

$$\min_{\mathbf{D}} \text{IRMSE}(\mathbf{D}, \boldsymbol{\theta}) = \min_{\mathbf{D}} \int_{\mathcal{X}} \int_{\mathcal{Z}} \sqrt{\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})} f(\mathbf{z}) d\mathbf{z} d\mathbf{x}. \quad (6)$$

This is a more meaningful criterion as it tries to minimize the expected volume of the confidence region of the predictions. We may generalize this criterion to

$$\min_{\mathbf{D}} \text{IRMSE}_k(\mathbf{D}, \boldsymbol{\theta}) = \min_{\mathbf{D}} \left[\int_{\mathcal{X}} \int_{\mathcal{Z}} \left\{ \sqrt{\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})} f(\mathbf{z}) \right\}^k / C_k d\mathbf{z} d\mathbf{x} \right]^{1/k} \quad (7)$$

for $k > 0$ and $C_k = \int_{\mathcal{Z}} f^k(\mathbf{z}) d\mathbf{z}$. The special case of $k = 2$ is of great interest as it is analytically tractable in some situations. Let $\text{IMSE} = \text{IRMSE}_2^2$. Thus,

$$\min_{\mathbf{D}} \text{IMSE}(\mathbf{D}, \boldsymbol{\theta}) = \min_{\mathbf{D}} \int_{\mathcal{X}} \int_{\mathcal{Z}} \text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta}) f^2(\mathbf{z}) / C_2 d\mathbf{z} d\mathbf{x}. \quad (8)$$

Interestingly, this is the same as the integrated mean squared error criterion in the literature

but with a weight function $f^2(\mathbf{z})$. In the case of uniform distributions, $f(\cdot)$ or $f^2(\cdot)$ doesn't make any difference, but for non-uniform distribution this does make a big difference. We will see later that $f^2(\cdot)$ gives the right scaling and provides meaningful solutions to the robust parameter design problem.

A major challenge of using the foregoing criteria is that they are functions of the *unknown* correlation parameters $\boldsymbol{\theta}$. One can minimize IRMSE (or IMSE) for a guessed value of $\boldsymbol{\theta}$, but the optimal design may not work well for another value of $\boldsymbol{\theta}$. A potential fix to overcome this problem is to first average the IRMSE over a prior distribution of $\boldsymbol{\theta}$ and find the design using

$$\min_{\mathbf{D}} \int \text{IRMSE}(\mathbf{D}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

This is a computationally intensive problem because \mathbf{R} is a function of $\boldsymbol{\theta}$ and thus, inverting \mathbf{R} and then integrating the IRMSE is time consuming. Moreover, this criterion doesn't work well in practice because $\text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}, \boldsymbol{\theta})$ increases with $\boldsymbol{\theta}$ and therefore, the criterion is dominated by the large values of $\boldsymbol{\theta}$. Sacks, Schiller and Welch (1989) proposed to overcome this problem by standardizing the criteria with respect to the optimal design obtained for a given value of $\boldsymbol{\theta}$. See Pratola et al. (2016) for a Bayesian version of this approach. Let Θ be a compact set containing the possible values of $\boldsymbol{\theta}$. Then, Sacks et al.'s approach is to find the design to maximize the minimum efficiency:

$$\max_{\mathbf{D}} \min_{\boldsymbol{\theta} \in \Theta} \frac{\text{IRMSE}(\mathbf{D}^*(\boldsymbol{\theta}), \boldsymbol{\theta})}{\text{IRMSE}(\mathbf{D}, \boldsymbol{\theta})}, \quad (9)$$

where $\mathbf{D}^*(\boldsymbol{\theta}) = \arg \min_{\mathbf{D}} \text{IRMSE}(\mathbf{D}, \boldsymbol{\theta})$. This criterion is extremely computationally intensive because one needs to find the optimal design for every possible value of $\boldsymbol{\theta} \in \Theta$, which is difficult in high dimensions. Sacks, Schiller and Welch (1989) tried to circumvent this problem by letting $\theta_i^x = \theta_j^z = \theta_0$ for all $i = 1, \dots, p$ and $j = 1, \dots, q$ and then taking a few discrete values of θ_0 . In our experience, this simplification results in designs having poor projections in subspaces of the factors, which is undesirable. One possible approach to improve the projections is to assign independent prior distributions for each of the unknown correlation parameters as in Joseph et al. (2015). However, this would make the computation of (9) very expensive. For example, even if we choose only three discrete values for each

correlation parameter, the total number of possible values in Θ would become 3^{p+q} which can be quite high when p and/or q is large.

Because of the foregoing difficulties, in the next section we will develop space-filling designs which are easy to compute and are model-robust. We will modify them so that they will perform well according to the criterion in (9).

3 SPACE-FILLING DESIGNS

Space-filling designs aim at filling the experimental region evenly with as few gaps as possible. These designs are robust to modeling choices and thus, are widely used as designs for computer experiments. Popular space-filling designs include Latin hypercube designs (McKay et al., 1979), distance-based designs such as maximin and minimax (Johnson et al., 1990), uniform designs (Fang and Wang, 1994), and several useful variants of them such as maximin Latin hypercube designs (Morris and Mitchell, 1995) and maximum projection designs (Joseph et al., 2015). See Joseph (2016) for a recent review of space-filling designs. However, these designs are developed for general purpose applications such as function approximation and not specifically for robustness experiments. As mentioned in the introduction, control-by-noise interactions are especially important for identifying robust settings. Therefore, we may hope to improve the performance of space-filling designs by improving their ability to estimate the control-by-noise interactions, possibly by sacrificing other not so important effects.

In the physical experiments' literature, there are mainly two classes of designs suitable for robustness experiments: cross arrays (Taguchi, 1987) and single arrays (Welch et al., 1990; Shoemaker et al., 1991). To develop single arrays, one needs to first quantify the importance of each effect (Bingham and Sitter, 2003; Wu and Zhu, 2003). This was not too difficult with the fractional factorial experiments because such designs usually have only two or three levels for each factor. In contrast, computer experiments have large number of levels for each factor and therefore, numerous effects are involved in the modeling. This makes the effect ordering a difficult task. Thus, cross arrays seems to be an easier and straightforward approach for computer experiments.

To develop cross arrays, we first need to choose an n_1 -run design \mathbf{D}_x for the control factors (known as control array) and another n_2 -run design \mathbf{D}_z for the noise factors (known as noise array). Cross array can then be obtained by repeating the noise array for each run of the control array, which will have a total of $n_1 n_2$ number of runs. We will denote the cross array by $\mathbf{D} = \mathbf{D}_x \times \mathbf{D}_z$. The suitability of a cross array for robustness experiments should be evident from its construction. We can estimate the effect of noise factors under each settings of the control factors in \mathbf{D}_x , which enables one to choose the control factor setting that makes the noise factors' effect on the response as small as possible. As shown in Wu and Hamada (2009) (see Theorem 11.1), if control factor effects are estimable from \mathbf{D}_x and noise factor effects from \mathbf{D}_z , then the two-factor interactions between control and noise factors are estimable and clear in \mathbf{D} .

The following result shows how to construct an optimal cross array that minimizes the IMSE criterion in (8). The proof is given in the Appendix.

Theorem 1. *If a product correlation is used between control and noise factors, then an IMSE-optimal cross array can be obtained by crossing an IMSE-optimal control array and an IMSE-optimal noise array.*

The IMSE optimal designs are closely related to the space-filling designs. For example, when $f(\mathbf{z}) = 1$, as $k \rightarrow \infty$ in (7),

$$\text{IRMSE}_\infty(\mathbf{D}) = \max_{\mathcal{X}} \max_{\mathcal{Z}} \text{MSE}(\mathbf{x}, \mathbf{z}; \mathbf{D}),$$

which is minimized by a minimax distance design when the correlations are small (Johnson et al. 1990). The assumption of small correlations is justifiable in the control factor space, but not in the noise factor space. This is mainly because the realistic ranges of noise factors are much smaller than the possible ranges of control factors and therefore, it is quite likely that the noise-response relationship can be adequately modeled using a smooth Gaussian process. Moreover, Noise factors have nonuniform distributions. We will discuss on how to modify the space-filling designs for the noise array in the next section.

A simple example can be used to illustrate why the cross arrays are useful in estimating control-by-noise interactions. Consider two control factors (x_1 and x_2) and two noise factors

$(z_1$ and $z_2)$. Suppose we choose a Maximin Latin hypercube design (MmLHD) with four runs for the control array and another MmLHD with five runs for the noise array. Then the cross array will have 20 runs. Their two-factor projections are shown in Figure 1. We can see that the projections are very poor in the control factor space and the noise factor space, but the projections are excellent on the control-by-noise factor space. Thus, we can obtain a good estimation of the control-by-noise interactions using this design. However, the estimation of the pure control or noise factor effects can be poor compared to using a 20-run MmLHD for the four factors.

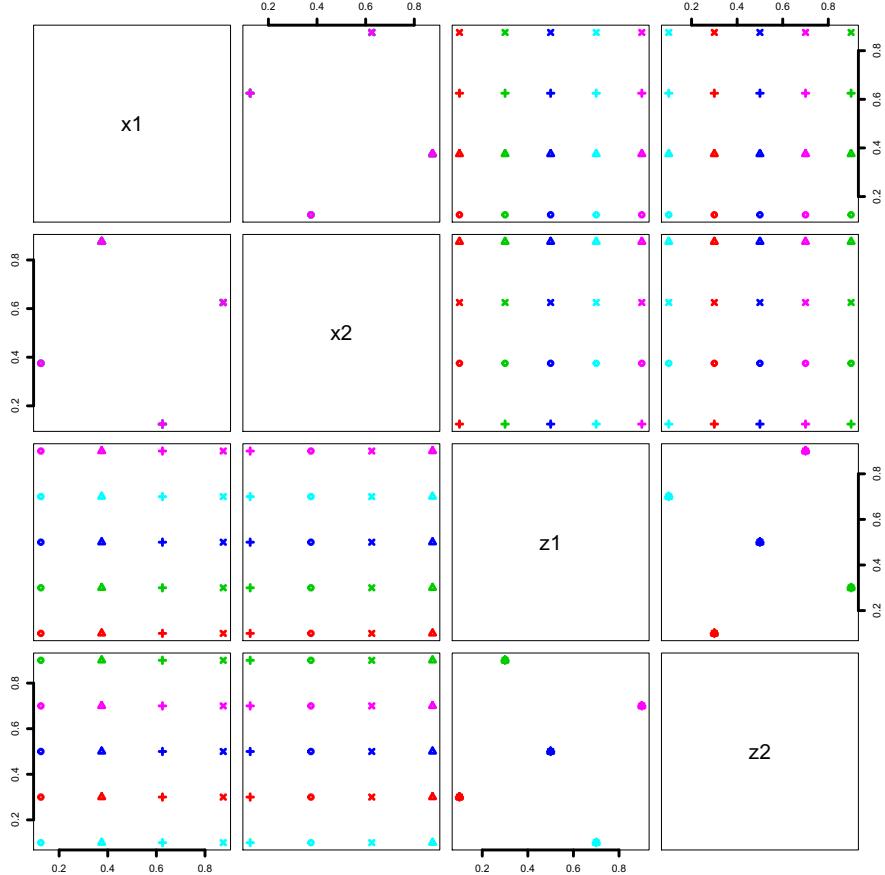


Figure 1: Two-dimensional projections of a cross array obtained by crossing a 4-run MmLHD for control factors and 5-run MmLHD for noise factors.

Although the cross array looks promising for robustness experiments, it has certain disadvantages for using in computer experiments. Each control factor level is replicated n_2

number of times and noise factor level n_1 number of times. Thus, the number of levels is much smaller than that of a comparable single array, which has $n_1 n_2$ number of levels for each factor. This can lead to poor estimation of nonlinear effects and higher order interactions in the control and noise factor spaces. This disadvantage of the cross array is amplified if there are only a few factors that are active. We propose an idea to overcome this disadvantage.

We can jitter each point in the cross array to increase the number of levels for each factor. The resulting design will still possess the good estimation ability of control-by-noise interactions because the response values observed over adjacent points are expected to be highly correlated. However, if the jittering radius is small, then there will be no improvement in the estimation of the pure control and noise factor effects. On the other hand, if the jittering radius is large, then the resulting design can lose its ability to efficiently estimate the control-by-noise interactions. So how much to jitter is a critical question and we have an intuitive solution for this. Let J_i be the cross product of the voronoi regions of the design points in the control and noise arrays corresponding to the i th point in the cross array. Now jitter the i th point within J_i . We will call the resulting design a *Jittered Cross Array* (JCA).

We can do better than a random jittering. Since our aim is to overcome the issue with projections, we can choose the points in J_i , $i = 1, \dots, n$, that will ensure good projections. We propose a sequential algorithm for doing this. Start from the center point $(.5, \dots, .5)$ and add one point at a time from J_i sequentially using the maximum projection (MaxPro) criterion (Joseph et al., 2015):

$$\mathbf{D}_i = \min_{\mathbf{u} \in J_i} \sum_{j=1}^{i-1} \frac{1}{\prod_{l=1}^p (u_l - \mathbf{D}_{jl})^2}, \quad (10)$$

for $i = 2, \dots, n$. This algorithm adds points sequentially in a greedy manner such that the i th point is as far as possible from the previously chosen points under the MaxPro criterion. Although the MaxPro criterion ensures that no two levels can be the same, the levels may not be equally spaced. So at the end we force them to be equally spaced, which can be easily done by ordering the levels. So the final design is like a Latin hypercube design (LHD), but with some clustering in the control and noise factor spaces. JCA shouldn't be confused with a cascading LHD (Handcock, 2007), which has clusters in the full-dimensional space and

doesn't have a crossed array structure.

The two-dimensional projections of the JCA for the previous example is shown in Figure 2. We can now see the 20 points in the $x_1 \times x_2$ and $z_1 \times z_2$ projections as opposed to only four and five points in the cross array. The projections in the control-by-noise spaces are still very good. We can also observe the clusters in the control (symbols) and noise (color) spaces, which shows that the cross array structure is approximately maintained. We will study the performance of these designs using simulated and real examples in a later section after deciding the optimal choice of noise array.

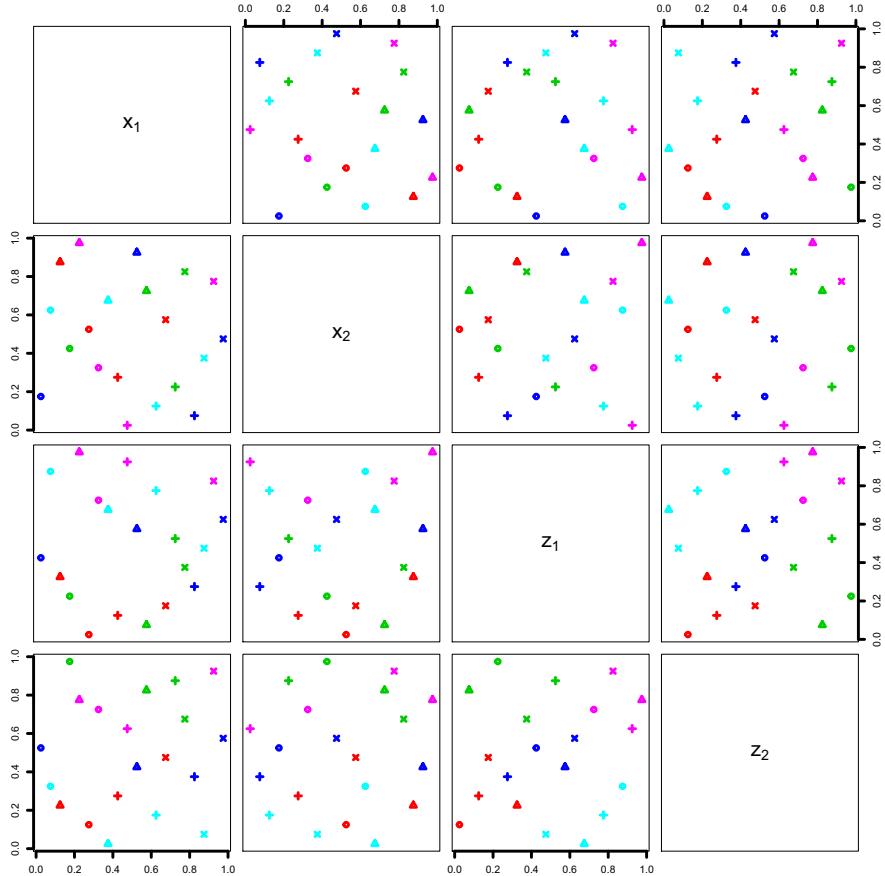


Figure 2: Two-dimensional projections of the jittered cross array in 20 runs with two control and two noise factors. The control factor levels in the cross array are coded by the plotting symbols and noise factor levels by color.

4 NOISE ARRAY

In this section we will discuss three possibilities for the choice of noise array. The first one is the most intuitive choice, but the latter two are better for robustness experiments.

4.1 Transformed Design

As mentioned earlier, the distribution of the noise factors are usually nonuniform. On the other hand, most space-filling designs are closely related to a uniform distribution as they try to spread out the points evenly in the experimental region. This suggests that we can possibly use the inverse probability transform method to transform the noise factor columns in a space-filling design to have the right distribution. More specifically, assume that the noise factors are independent. Let $F_l(z_l)$ be the distribution function of z_l , $l = 1, \dots, q$. Then it can be shown that if the design $\{\mathbf{z}_1, \dots, \mathbf{z}_q\}$ minimizes the discrepancy from a uniform distribution, then $\{F_1^{-1}(\mathbf{z}_1), \dots, F_q^{-1}(\mathbf{z}_q)\}$ minimizes the F -discrepancy (Fang and Wang, 1994, p. 21), where the transformation is applied element-wise. We will call this design a *transformed design*. Note that in computer experiments, we don't need to worry about the number of noise levels or the spacing among the levels as in the case of physical experiments (Del Castillo et al., 2007).

The independence assumption is crucial for the above simplification. This assumption can be easily relaxed when the noise factors follows a multivariate normal distribution: $N(\mathbf{0}, \Sigma_z)$. We can first find the space-filling design assuming independence and then transform using $\Sigma^{1/2}\Phi^{-1}(\mathbf{z}^i)$, where $\Phi(\cdot)$ is the distribution function of a standard normal variable and \mathbf{z}^i is the i th row of the space-filling design. If the distribution is not normal, then the design can be found using the idea of support points (Mak and Joseph, 2017b).

For illustration, let z be a normal distribution with mean 0.5 and standard deviation $\sigma = 1/6$. These values are chosen so that the $\pm 3\sigma$ limits of z coincide with $[0, 1]$. Suppose we use $n = 10$ and a Gaussian correlation function: $R(z_i - z_j) = \exp\{-\theta(z_i - z_j)^2\}$. As shown in (Fang and Wang, 1994, p.19), $\mathbf{d}_0 = \{0.5/n, 1.5/n, \dots, (n - .5)/n\}$ minimizes the discrepancy from a uniform distribution. So the desired design can be obtained as $F^{-1}(\mathbf{d}_0)$.

Figure 3 plots the weighted root mean squared error

$$WRMSE(\mathbf{x}, \mathbf{z}) = \sqrt{MSE(\mathbf{x}, \mathbf{z})} f(\mathbf{z}), \quad (11)$$

for $\theta = 10$ and $\theta = 1000$. The design points are shown as crosses in the same plots. We can see that the points are pulled towards the center as one would expect for a normal distribution. However, although the transformation seems to balance the WRMSE well throughout the region when $\theta = 1000$, it seems to be too high in the tail regions when $\theta = 10$. Thus, the benefit of using a transformed design seems to depend on the smoothness of the underlying response function. If the function is wiggly, then transformation will work great, but if the function is smooth, the transformation may do more harm than good. As mentioned before, the external noise factors usually have a smooth relationship with the response. Thus, θ is expected to be small in the Gaussian correlation function. In summary, a transformed design does not seem to be a good choice for the noise array and we need to look for other alternatives.

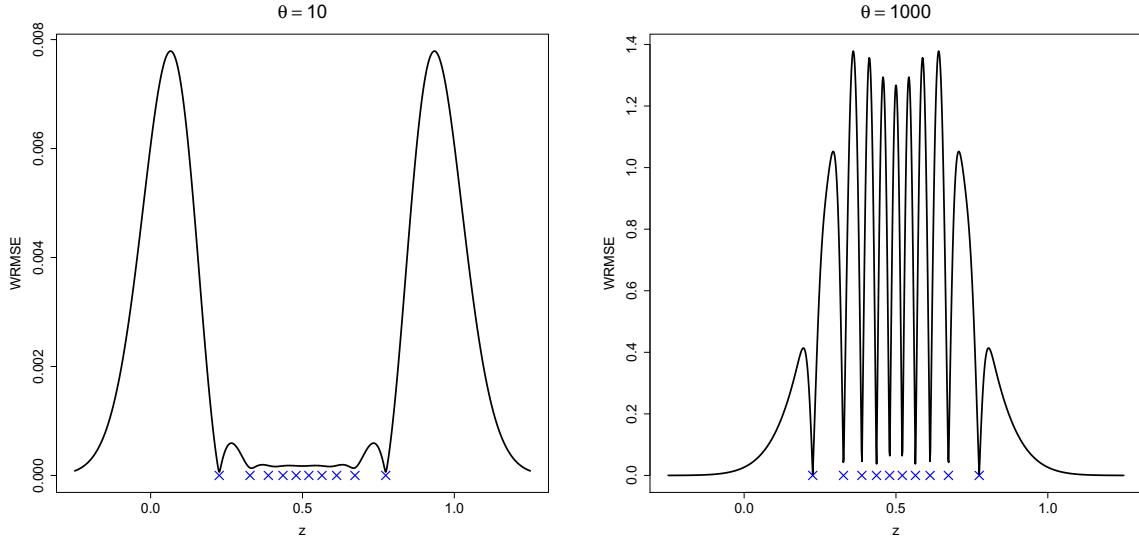


Figure 3: Plot of weighted root mean squared error against the noise factor for a 10-point transformed design using $N(.5, \sigma)$ for $\theta = 10$ (left) and $\theta = 1000$ (right).

4.2 Hybrid Design

We can find the design that minimizes IRMSE in (6) for a given value of θ or even better by maximizing the efficiency in (9). However, as mentioned before, this optimization is hard to perform, especially in high dimensions. We propose a simple idea to overcome this problem. We will find the optimal design for one factor, which is easier. The optimal design can be viewed as a transformation of a uniform design. Now we use this optimal transformation on each column of the noise factors of a space-filling design. Since the final design is obtained using a combination of space-filling and model-based optimal design criteria, we will call the design a *hybrid design*.

To fix the idea, consider a single noise factor z . Let $\mathbf{d}^* = \{z_1^*, \dots, z_n^*\}$ be the optimal design obtained using the model-based criterion

$$\max_{\mathbf{d}} \min_{\theta \in \Theta} \frac{\text{IRMSE}(\mathbf{d}^*(\theta), \theta)}{\text{IRMSE}(\mathbf{d}, \theta)},$$

where

$$\mathbf{d}^*(\theta) = \arg \min_{\mathbf{d}} \int_{\mathcal{Z}} \sqrt{\text{MSE}(z; \mathbf{d}, \theta)} f(z) dz.$$

We want to emphasize that these optimizations are computationally much simpler than those of (9) and (6) which use the full factor space. Let $\{u_i^* = (i - .5)/n, i = 1, \dots, n\}$ be the uniform design points in $[0, 1]$. It is easy to show that the optimal design points are distinct, that is $z_i^* \neq z_j^*$ for $i \neq j$. Therefore, there exists a one-to-one transformation:

$$z_i^* = T(u_i^*), \text{ for } i = 1, \dots, n. \quad (12)$$

For the case of multiple factors, let \mathbf{U}_z be a space-filling design with levels $\{u_1^*, \dots, u_n^*\}$ for each of the noise factor. Then, the hybrid design can be obtained as

$$\mathbf{D}_z^* = T(\mathbf{U}_z), \quad (13)$$

where the transformation $T(\cdot)$ is applied on each element of \mathbf{U}_z .

Consider again the same example of the previous subsection with $z \sim N(.5, 1/6)$ and

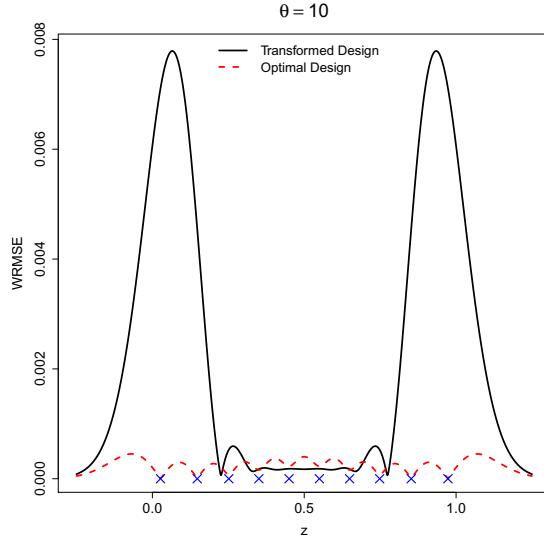


Figure 4: Plot of weighted mean squared error against the noise factor for a 10-point transformed design and optimal design with $\theta = 10$. The optimal design points are shown as crosses.

$\theta = 10$. The WRMSE for the 10-run optimal design is plotted in Figure 4. We can see that the optimal design performs much better than the transformed design. However, this improvement is not realizable in practice because we never know the true value of θ before the experiment. We need to choose a robust value of θ using (9).

Consider a set $\Theta = \{5, 10, 20, 30\}$ with $z \sim N(.5, 1/6)$ and $n = 50$. We first find the optimal designs for each of the four values of $\theta \in \Theta$. A histogram of the points for one case is shown in Figure 5. We can see that the IRMSE-optimal design points are slightly more dispersed than the noise distribution. The efficiencies of the four designs are computed using (9) and are plotted over θ in Figure 6. We can see that the optimal design found based on smaller values of θ perform poorly for larger values of θ . On the other hand, the optimal designs found using larger values of θ perform not so poorly for smaller values of θ . This suggests that we should find the set of probable values of θ and use the largest value in that set to generate the optimal design.

Guessing the largest possible value of θ can be challenging in a practical problem. Moreover, the numerical inaccuracies in computing IRMSE increases with n and the optimization becomes harder and unstable. Due to these difficulties, in the next subsection, we will try

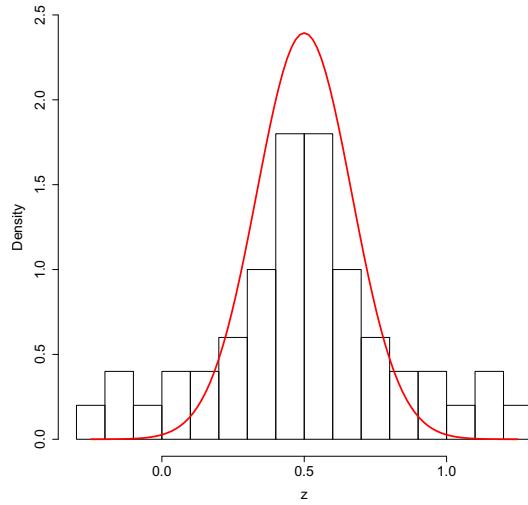


Figure 5: Histogram of the IRMSE-optimal design obtained for $n = 50$ and $\theta = 30$. The density of the noise distribution is also shown.

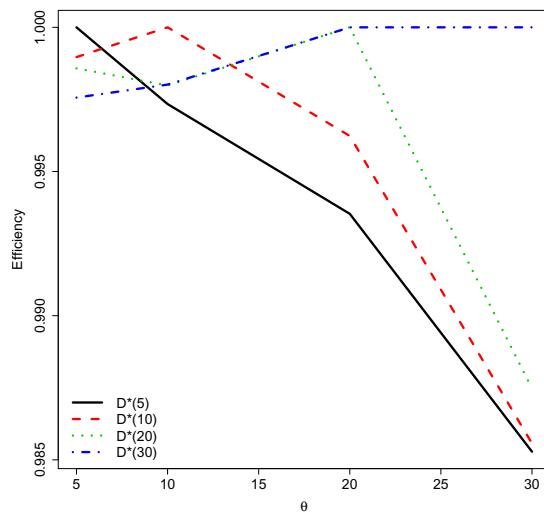


Figure 6: Efficiency of the optimal design obtained for a given value of θ plotted against different values of θ .

to identify an approximate optimal design that is easier to use in practical applications.

4.3 Double Transformed Design

Let $\mathbf{d}^* = \{z_1^*, \dots, z_n^*\}$ be the optimal set of n points that minimizes the IRMSE in (6) for a given value of θ . The empirical distribution function of this point set is given by $F_n(z) = 1/n \sum_{i=1}^n I(z_i^* < z)$, where $I(\cdot)$ is the indicator function. Our aim is to understand the limiting distribution of $F_n(z)$ as $n \rightarrow \infty$, which we denote by $\tilde{F}(z)$. This is not an easy problem because there is no explicit solution for the optimal design. Moreover, IRMSE is a complex function of the design points. Therefore, we will make use of an existing result on optimal designs for uniform distributions to get an idea of the limiting distribution.

For uniformly distributed variables, Dette and Pepelyshev (2010) showed that a beta distribution with density $b(z; \alpha, \alpha)$ for $\alpha \in [0.5, 1]$ is optimal for a reciprocal distance criterion which can be viewed as a surrogate for (6). See also Zhigljavsky et al. (2010) for a rigorous justification of this result. We know that if $F(\cdot)$ is the distribution function of z , then $F(z) \sim U(0, 1)$. Thus, by using change of variables, the optimal density of the design can be obtained as

$$\begin{aligned}\tilde{f}(z) &= b(F(z); \alpha, \alpha) f(z) \\ &= \frac{\Gamma(2\alpha)}{\Gamma^2(\alpha)} \frac{f(z)}{\{F(z)[1 - F(z)]\}^{1-\alpha}}.\end{aligned}\tag{14}$$

Let $B_\alpha(z) = \int_0^z b(u; \alpha, \alpha) du$ be the distribution function of the beta distribution. Then, the asymptotic distribution function of the optimal design is given by

$$\tilde{F}(z) = B_\alpha(F(z)).\tag{15}$$

Thus, if \mathbf{d}_0 denotes the uniform design, then an approximation to the optimal design can be obtained as

$$\mathbf{d}^* = F^{-1}(B_\alpha^{-1}(\mathbf{d}_0)).\tag{16}$$

To distinguish from the previous transformed design $F^{-1}(\mathbf{d}_0)$, we will call this the *double transformed design*.

What should be the value of α ? If θ is large in the Gaussian correlation function, we should use $\alpha = 1$, which leads to the transformed design. We have seen in Figure 3 that the transformed design indeed works well with large θ . However, as mentioned earlier, we are more interested in small values of θ . Dette and Pepelyshev (2010) recommended using $\alpha = 1/2$, which is the limiting distribution of a D -optimal design for large degree polynomial regression. Theorem 10.1 in Fasshauer and McCourt (2016) shows that as the correlation parameter $\theta \rightarrow 0$, the Gaussian process predictor tends to a high degree polynomial interpolator and therefore, it makes sense to use an α value close to 1/2. The design points, in this case, are the same as Chebyshev nodes, which possess minimax optimality properties for polynomial interpolation (Trefethen, 2013). However, as discussed towards the end of previous subsection, the optimal design based on a small value of θ may work poorly for large values of θ . Thus, intuitively, a value of α slightly larger than 1/2 such as 2/3 or 3/4 might be a more robust choice. We investigate this more carefully below for the case of a normal distribution.

It is easy to show that $\text{MSE}(z; \mathbf{d}) = 1 - \mathbf{r}(z)' \mathbf{R}^{-1} \mathbf{r}(z) \leq 1 - R^2(z - Q(z, \mathbf{d}))$, where $Q(z, \mathbf{d})$ is the closest point in \mathbf{d} to z . Let

$$\overline{\text{IRMSE}}(\mathbf{d}) = \int_{\mathcal{Z}} \sqrt{1 - R^2(z - Q(z, \mathbf{d}))} f(z) dz,$$

which is an upper bound of $\text{IRMSE}(\mathbf{d})$. For Gaussian correlation function,

$$\begin{aligned} \overline{\text{IRMSE}}(\mathbf{d}) &= \int_{\mathcal{Z}} \sqrt{1 - \exp\{-2\theta(z - Q(z, \mathbf{d}))^2\}} f(z) dz \\ &\approx \sqrt{2\theta} \int_{\mathcal{Z}} |z - Q(z, \mathbf{d})| f(z) dz, \end{aligned} \tag{17}$$

where the approximation is valid for large n . Zador (1982) has shown that the design that minimizes $\int |z - Q(z, \mathbf{d})|^k f(z) dz$ has an asymptotic distribution proportional to $f^{1/(1+k)}(z)$. This implies that the asymptotic distribution of \mathbf{d} that minimizes $\overline{\text{IRMSE}}(\mathbf{d})$ should be proportional to $\sqrt{f(z)}$. Now consider the case of a normal distribution $f(z) = \phi(z; .5, \sigma)$. Based on the simple approximation given by Bell (2015), $F(z)[1 - F(z)] \approx f^{4/\pi}(z)$. Substituting this approximation in (14), we obtain $\tilde{f}(z) \approx \Gamma(2\alpha)/\Gamma^2(\alpha) \{f(z)\}^{1-4(1-\alpha)/\pi}$. This

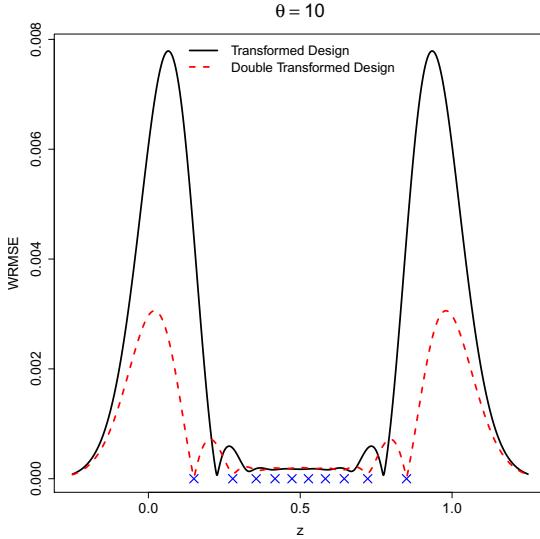


Figure 7: Plot of weighted mean squared error against the noise factor for a 10-point transformed design using $N(.5, \sigma)$, and double transformed design using (16) with $\alpha = 2/3$ for $\theta = 10$.

will be proportional to $\sqrt{f(z)}$ if $\alpha = 1 - \pi/8 \approx 0.607$. Similar exercise using IMSE gives $\alpha = 1 - \pi/12 \approx 0.738$. Based on these values, we choose $\alpha = 2/3 \approx (.607 + .738)/2$. Interestingly, in a totally different problem setting of searching for the maximum of a continuous function using non-adaptive algorithms, Al-Mharmah and Calvin (1996) showed that the same beta density with $\alpha = 2/3$ is optimal.

Consider the previous example with $z \sim N(.5, 1/6)$. The WRMSE for the double transformed design using (16) with $\alpha = 2/3$ is shown in Figure 7. We can see that WRMSE for the double transformed design is much smaller than that using the transformed design, but not as good as the optimal design in Figure 4.

The left panel of Figure 8 compares the noise distribution with the asymptotic optimal density in (14). We can see that the density for the optimal design is more dispersed than the original noise distribution. The right panel of Figure 8 shows the optimal density when the noise distribution is truncated to $[0, 1]$. This density has three modes: one at the center and two at the boundaries. This is an interesting result, because Taguchi (1987) has recommended using three levels for the noise factor, one at the mean and two at the extremes. The result in Figure 8 can be viewed as an extension of this three-level design

for physical experiment to an n -level design for computer experiment, where projections are important.

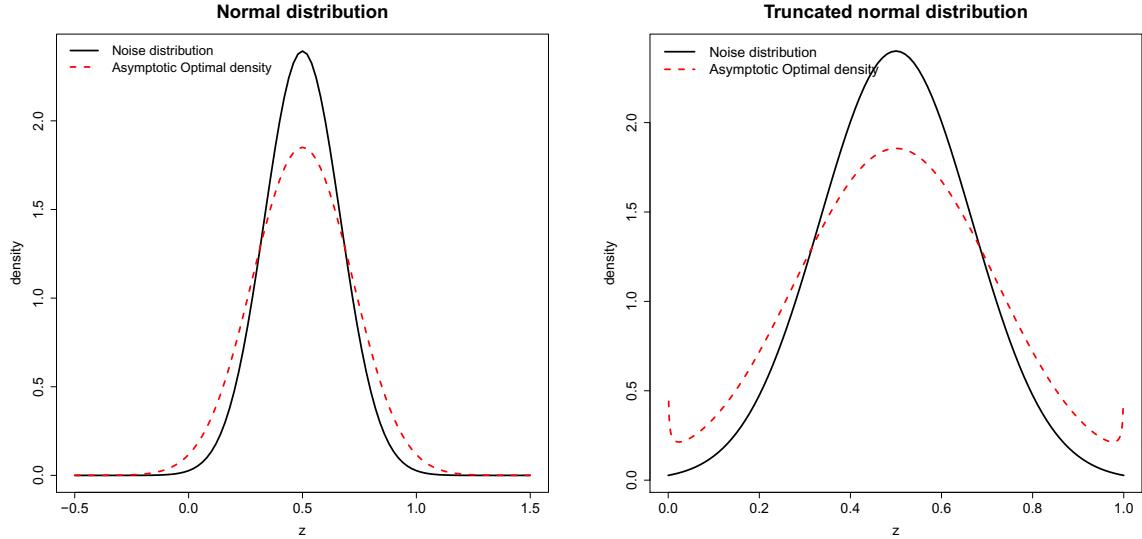


Figure 8: Comparison of $N(.5, \sigma)$ (left) and the asymptotic optimal density in (14) with $\alpha = 2/3$. The right panel shows the densities when the normal distribution is truncated in $[0, 1]$.

A similar investigation using the initial IMSE criterion in (5) shows that $\alpha \approx 0.476$ is the optimal choice. However, this makes the optimal density quite dispersed and places points in very low probability regions. For example, when $n = 100$, the points can be as far as $\pm 3.95\sigma$ from the center. On the other hand, $\alpha = 2/3$ places points within $\pm 3.25\sigma$ from the center, which looks more reasonable. This is why we feel the IRMSE criterion in (6) or the IMSE criterion in (8) is more meaningful than the IMSE criterion in (5). This was also verified using the prediction performance on some test cases.

5 Factors with Internal Noise

There are some factors whose nominal values can be controlled, but they can vary around their nominal values. Such factors are said to have *internal noise*. Examples include, part-to-part variability within their manufacturing tolerances and process parameter variability around its target. On the other hand, *external noise* factors are completely uncontrollable

including their nominal values. Examples of external noise factors include user conditions, incoming raw material properties, etc. In this section we will propose methods for designing experiments with internal noise factors. One may wonder why we need to consider these factors differently from the external noise factors. Why not just merge them with the external noise factors and use the techniques described in the previous two sections? The reason is that we don't need to vary the internal noise factors in the experiment! They can be easily introduced at the modeling stage. Thus, internal noise factors can be ignored at the design stage although this may not be the “optimal” approach. This topic of designing experiments with internal noise factors has received scant attention in the literature except possibly for the work of Kang and Joseph (2009) in the case of physical experiments.

A factor with internal noise can be represented as $X = x + e$, where the nominal value x is controllable and the internal noise e is uncontrollable. Here we have used additive noise, but the case of multiplicative noise can be handled similarly. Suppose we have estimated the relationship with the nominal values of the factors (\mathbf{x}) and external noise factors (\mathbf{z}): $\hat{g}(\mathbf{x}, \mathbf{z})$, then we can easily obtain the relationship with the internal noise as $\hat{g}(\mathbf{x} + \mathbf{e}, \mathbf{z})$. This is why we don't need to vary \mathbf{e} in the experiment.

Since a factor with internal noise is both a “control” and a “noise” factor, it make sense to cross this factor with the other factors in the experiment. Thus, if \mathbf{D}_X denotes the design for the factors with internal noise, then we can obtain the cross array using $\mathbf{D}_x \times \mathbf{D}_z \times \mathbf{D}_X$. From this we can obtain the jittered cross array using the same algorithm discussed earlier. Now, we only need to decide on how to choose the levels for a factor with internal noise.

As before let's assume x to follow a uniform distribution in $[0, 1]$ and e to have a noise distribution with density $f_e(e)$. Because the internal noise factor is the result of not controlling the process well, it is mostly going to have a normal distribution. So let $e \sim N(0, \sigma_e)$. The optimal design $\mathbf{d}^* = \{x_1^*, \dots, x_n^*\}$ can be obtained by minimizing

$$\text{IRMSE}(\mathbf{d}, \theta) = \int_0^1 \int_{-\infty}^{\infty} \sqrt{1 - \mathbf{r}(x + e)' \mathbf{R}^{-1} \mathbf{r}(x + e)} \phi(e; 0, \sigma_e) d e d x,$$

where $\phi(e; 0, \sigma_e)$ denote the density of a normal distribution with mean 0 and standard deviation σ_e . Here it is better to consider the IMSE criterion in (8) because an explicit

expression for the integral can be obtained under a Gaussian correlation function, $R(h) = \exp(-\theta h^2)$. Thus,

$$\text{IMSE}(\mathbf{d}, \theta) = \int_0^1 \int_{-\infty}^{\infty} \{1 - \mathbf{r}(x+e)' \mathbf{R}^{-1} \mathbf{r}(x+e)\} \phi^2(e; 0, \sigma_e) / C_2 d e d x \quad (18)$$

$$= \frac{1}{2\sigma_e \sqrt{\pi} C_2} \left[1 - \int_0^1 \text{tr} \{ \mathbf{R}^{-1} \mathbf{A}(x) \} dx \right], \quad (19)$$

where the ij th element of $\mathbf{A}(x)$ is given by

$$\mathbf{A}_{ij}(x) = \frac{1}{\sqrt{1 + 2\theta\sigma_e^2}} \exp \left\{ -\frac{2\theta}{1 + 2\theta\sigma_e^2} \left(x - \frac{x_i + x_j}{2} \right)^2 \right\} \exp \left\{ -\frac{\theta}{2} (x_i - x_j)^2 \right\},$$

for $i, j = 1, \dots, n$. The integration with respect to x can also be done explicitly to obtain

$$\text{IMSE}(\mathbf{d}, \theta) = \frac{1}{2\sigma_e \sqrt{\pi} C_2} [1 - \text{tr} \{ \mathbf{R}^{-1} \bar{\mathbf{A}} \}], \quad (20)$$

where

$$\bar{\mathbf{A}}_{ij} = \frac{\sqrt{\pi}}{\sqrt{2\theta}} \left\{ \Phi \left(\frac{\sqrt{\theta}(2 - x_i - x_j)}{\sqrt{1 + 2\theta\sigma_e^2}} \right) - \Phi \left(\frac{-\sqrt{\theta}(x_i + x_j)}{\sqrt{1 + 2\theta\sigma_e^2}} \right) \right\} \exp \left\{ -\frac{\theta}{2} (x_i - x_j)^2 \right\},$$

where $\Phi(\cdot)$ is the standard normal distribution function. Similar explicit expressions could have been obtained in the previous section as well for the external noise factors, but we didn't do it because of numerical issues. We found the formula in (20) to be quite vulnerable to numerical issue when \mathbf{R} is nearly singular, which happens when θ is small. For the external noise factors, we need to consider small values of θ because they are expected to have a smooth relationship with y . On the other hand, the relationship with x can be quite rough and therefore, here we should use large values of θ which doesn't lead to numerical problems.

As an example, consider a factor with internal noise distribution $e \sim N(0, \sigma_e)$. Let $\sigma_e = 1/12$ and $\theta = 50$ in the Gaussian correlation function. A 10-point optimal design is obtained by numerically minimizing (20). Figure 9 plots the expected mean squared error for the 10-point uniform design and the optimal design. We can see that the optimal design is almost equally spaced but with points placed at the boundaries. Our simulations show

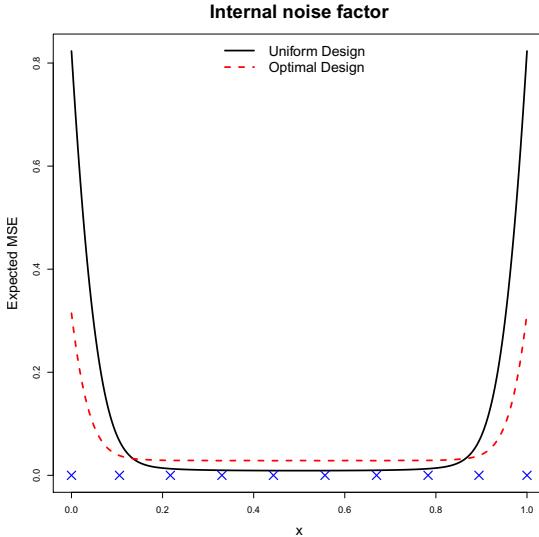


Figure 9: Comparison of expected MSE for the 10-point uniform design (solid) and optimal design (dashed) when $\sigma_e = 1/12$ and $\theta = 50$. The optimal design points are shown as crosses.

that $\{0, 1/(n-1), 2/(n-1), \dots, 1\}$ is close to optimal, which is only a slight change from the uniform design.

6 EXAMPLES

6.1 A Simulated Example

Consider a simple example with one control and four external noise factors. Let

$$y = \sum_{i=1}^4 \beta_i(x - \gamma_i) z_i^2 e^{-(x-\gamma_i)^2},$$

where $z_1, z_2 \sim^{iid} N(0, 1)$. We constructed a cross array using $n_1 = 3$ equally spaced levels for the control factor and a minimax distance design with $n_2 = 9$ runs for the noise factors (Mak and Joseph, 2017a). A minimax design is chosen for the noise array because it minimizes the size of the voronoi regions and thus, the jittering will have minimum impact on the cross array structure. From this a JCA is obtained using the sequential MaxPro algorithm described in Section 3. For comparison, we also constructed a MaxProLHD in 27 runs. We consider

two versions of these two designs using transformed noise array and double transformed noise array. This gives four designs: transformed MaxPro (TrMaxPro), double-transformed MaxPro (DTMaxPro), transformed JCA (TrJCA), and double-transformed JCA (DTJCA). We simulated 100 cases by randomly sampling β_i 's and γ_i 's from $U(0, 1)$. The designs were also generated 100 times to account for the small variations in the design algorithm. An R code for this simulation is provided in the supplementary materials associated with this paper.

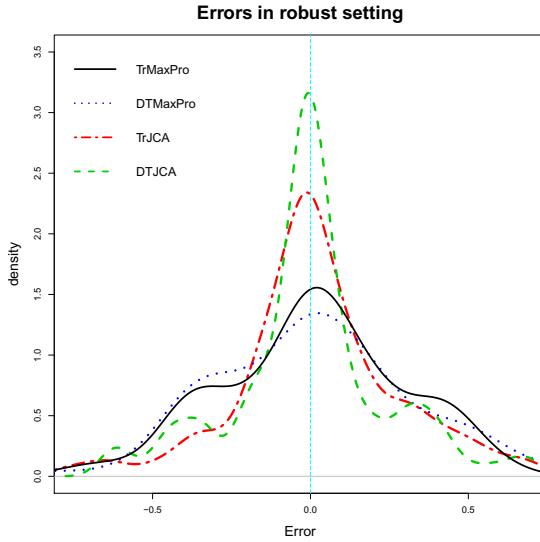


Figure 10: Density plots of errors in the robust settings based on the four designs in the simulation example.

Suppose our aim is to minimize the variance of the response due to the noise factors. First we fit kriging models using the data generated by each of the four designs. Then we computed the true robust setting of the control factor (x^*) by minimizing the variance and also the robust settings obtained from the fitted models based on the four designs (x_1^*, x_2^*, x_3^* , and x_4^*). Figure 10 shows the density plots of the errors $x_i^* - x^*$, for $i = 1, 2, 3, 4$. We can see that the DTJCA gives the best performance followed by TrJCA.

6.2 A Real Example

Computer experiments with noise factors are quite common for the simulations conducted at The Procter & Gamble Company. The specific example we will use in our study involves a manufacturing packing line. The example has been slightly modified for the benefit of simplicity and to prevent disclosure of any potential sensitive information. A computer simulator was developed for one critical transformation of the packing line involving transport of the package for product fill. A computer experiment with nine input factors was performed and an emulator was built. In this study, we use this emulator for investigating the robustness. Variables x_1, x_2, x_3, x_4, x_5 , and x_6 are process variables such as speed of the line and dimensions of the puck that transports the package that will remain fixed or easy to control once they are chosen and therefore are defined as control factors. Variables z_1 , z_2 , and z_3 are material properties of the packaging component such as density and modulus which are defined as noise factors given that there is variation in normal production of the material supplier. The output response from the computer simulation is the amount of deflection of the package (deviation from a vertical orientation) after the puck, which is holding the package, comes to a stop on the packing line for product filling. Figure 11 provides an illustration of dispensing the product into the package, which is represented by the vertical cylinder. The base, which holds the cylinder, is the puck that transports the package through the packing line. The illustration on the left represents no deflection which is favorable, while the illustration on the right represents large deflection which can cause quality issues on the packing line like product spill. A quality issue will require the packing line to be shut down and can have a significant impact on throughput. The objective of this study is to find the settings for the six control factors that are robust to the variation of the three noise factors. From historical data, the noise factors are found to be approximately normally distributed. After re-scaling, we let $z_i \sim^{iid} N(.5, \sigma)$ for $i = 1, 2, 3$ with $\sigma = 1/6$.

First we generated a minimax design with $n_1 = 2 \times 6 + 1 = 13$ runs for the six control factors and another minimax design with $n_2 = 2 \times 3 + 1 = 7$ runs for the three noise factors. The JCA in 91 runs is thus obtained using the sequential MaxPro algorithm and then performed the double transformation on the noise factor columns using (16). We

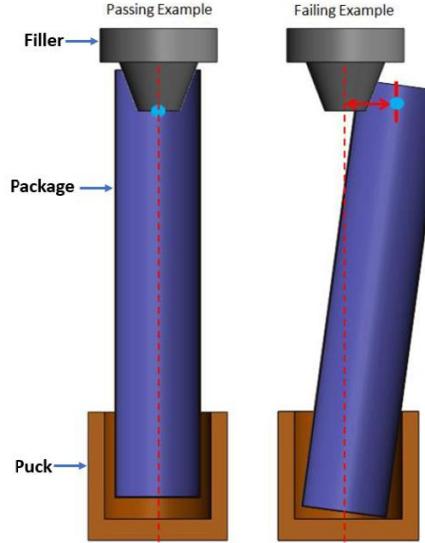


Figure 11: Illustration of the package deflection during product filling. Large deflections such as the one shown on the right side can cause product spill.

also generated a MaxProLHD in 91 runs for comparison and transformed using the noise distribution. Kriging models were fitted to the data generated from the two designs. We found the prediction errors from the two fitted models to be close, but there was some major differences in the estimation of control-by-noise interactions. Figure 12 shows the interaction between x_2 and z_3 , which is the most significant interaction in this experiment. Clearly, the new double transformed JCA did a much better job in accurately estimating the interaction than the existing transformed MaxProLHD.

7 CONCLUSIONS

In this paper we have proposed space-filling designs that are suitable for identifying robust settings using computer experiments. The key idea was to jitter the points in a cross array using the maximum projection criterion. We called the new design *Jittered Cross Array*. We have also proposed how to optimally choose the noise array. The most intuitive way to construct the noise array is to transform the columns of the array using the inverse cumulative distribution function of the noise factors. This will pull the design points to the high probability region of the noise distribution. However, we found this “pulling effect”

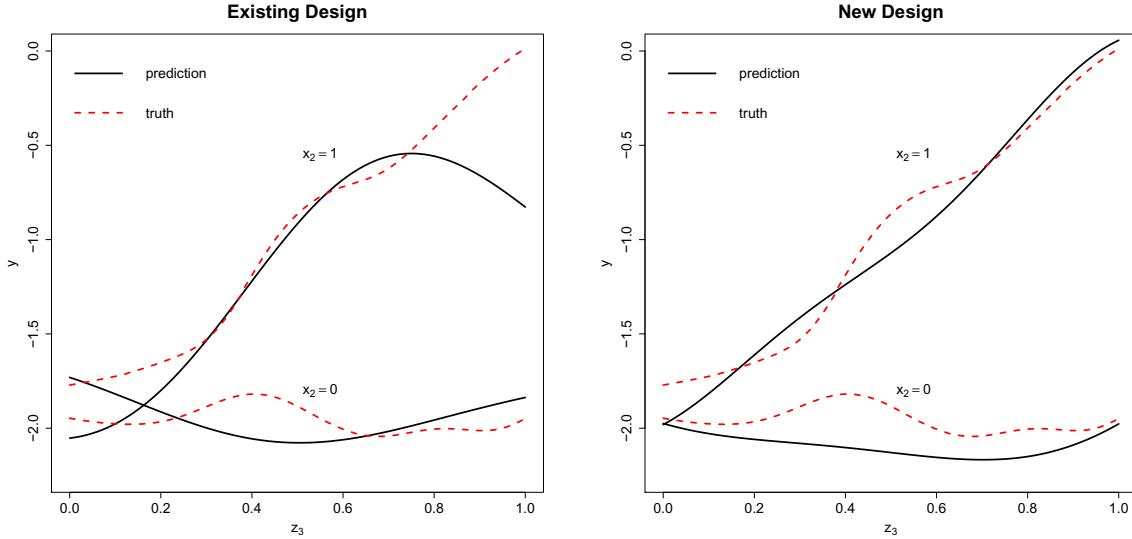


Figure 12: Interaction plot of x_2 against z_3 in the real example. The left panel shows the interaction obtained using the existing transformed MaxProLHD and the right panel using the new double transformed JCA.

to be too extreme. This was mainly because the response is usually a smooth function of the noise factors and to precisely estimate a smooth function it is desirable to push the points outward from the center. The optimal design balances this “pulling” and “pushing” effects. We found that pushing the points uniformly distributed in the unit interval using a $Beta(2/3, 2/3)$ distribution before applying the inverse probability transform to be close to asymptotically optimal for normally distributed noise variables. We have also proposed model-based methods to obtain the optimal transformation for any noise distribution, but it requires specification of the upper bound of certain correlation parameters.

A disadvantage of the jittered cross array is that the total number of runs can become prohibitively large when the number of control and noise factors is large. One way to overcome this is to use single arrays. We believe that developing such arrays for computer experiments will be a good direction for future research.

SUPPLEMENTARY MATERIALS

R codes and datasets: The R codes for the simulation in Section 6.1 and the datasets for the example in Section 6.2 can be downloaded as a .zip file.

ACKNOWLEDGMENTS

This research is supported by a U.S. National Science Foundation grant DMS-1712642 and a U.S. Army Research Office grant W911NF-17-1-0007.

APPENDIX: Proof of Theorem 1

Consider a production correlation $R(\mathbf{x}_i - \mathbf{x}_j, \mathbf{z}_i - \mathbf{z}_j) = R_x(\mathbf{x}_i - \mathbf{x}_j)R_z(\mathbf{z}_i - \mathbf{z}_j)$. Now, if $\mathbf{D} = \mathbf{D}_x \times \mathbf{D}_z$, then $\mathbf{R} = \mathbf{R}_x \otimes \mathbf{R}_z$ and $\mathbf{r}(\mathbf{x}, \mathbf{z}) = \mathbf{r}_x(\mathbf{x}) \otimes \mathbf{r}_z(\mathbf{z})$, where \otimes denotes Kronecker product (see, for example, Hung et al. 2015). Then, by using the properties of Kronecker products

$$\begin{aligned}
\mathbf{r}(\mathbf{x}, \mathbf{z})' \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}, \mathbf{z}) &= (\mathbf{r}_x(\mathbf{x}) \otimes \mathbf{r}_z(\mathbf{z}))' (\mathbf{R}_x \otimes \mathbf{R}_z)^{-1} \mathbf{r}_x(\mathbf{x}) \otimes \mathbf{r}_z(\mathbf{z}) \\
&= (\mathbf{r}_x(\mathbf{x})' \otimes \mathbf{r}_z(\mathbf{z})') (\mathbf{R}_x^{-1} \otimes \mathbf{R}_z^{-1}) \mathbf{r}_x(\mathbf{x}) \otimes \mathbf{r}_z(\mathbf{z}) \\
&= (\mathbf{r}_x(\mathbf{x})' \mathbf{R}_x^{-1} \otimes \mathbf{r}_z(\mathbf{z})' \mathbf{R}_z^{-1}) \mathbf{r}_x(\mathbf{x}) \otimes \mathbf{r}_z(\mathbf{z}) \\
&= (\mathbf{r}_x(\mathbf{x})' \mathbf{R}_x^{-1} \mathbf{r}_x(\mathbf{x})) \otimes (\mathbf{r}_z(\mathbf{z})' \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z})) \\
&= \mathbf{r}_x(\mathbf{x})' \mathbf{R}_x^{-1} \mathbf{r}_x(\mathbf{x}) \mathbf{r}_z(\mathbf{z})' \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}).
\end{aligned}$$

Thus,

$$\begin{aligned}
\text{IMSE}(\mathbf{D}) &= \int_{\mathcal{X}} \int_{\mathcal{Z}} \{1 - \mathbf{r}_x(\mathbf{x})' \mathbf{R}_x^{-1} \mathbf{r}_x(\mathbf{x}) \mathbf{r}_z(\mathbf{z})' \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z})\} f^2(\mathbf{z}) / C_2 d\mathbf{z} d\mathbf{x} \\
&= 1 - \int_{\mathcal{X}} \mathbf{r}_x(\mathbf{x})' \mathbf{R}_x^{-1} \mathbf{r}_x(\mathbf{x}) d\mathbf{x} \int_{\mathcal{Z}} \mathbf{r}_z(\mathbf{z})' \mathbf{R}_z^{-1} \mathbf{r}_z(\mathbf{z}) f^2(\mathbf{z}) / C_2 d\mathbf{z} \\
&= 1 - (1 - \text{IMSE}(\mathbf{D}_x))(1 - \text{IMSE}(\mathbf{D}_z)).
\end{aligned}$$

Thus, $\min_{\mathbf{D}} \text{IMSE}(\mathbf{D}) = 1 - (1 - \min_{\mathbf{D}_x} \text{IMSE}(\mathbf{D}_x))(1 - \min_{\mathbf{D}_z} \text{IMSE}(\mathbf{D}_z))$.

References

Al-Mharmah, H., and Calvin, J. M. (1996), “Optimal Random Non-Adaptive Algorithm for Global Optimization of Brownian Motion,” *Journal of Global Optimization*, 8, 81–90.

Apley, D. W., and Kim, J. (2011), “A Cautious Approach to Robust Design with Model Parameter Uncertainty,” *IIE Transactions*, 43(7), 471–482.

Apley, D. W., Liu, J., and Chen, W. (2006), “Understanding the Effects of Model Uncertainty in Robust Design with Computer Experiments,” *Journal of Mechanical Design*, 128(4), 945–958.

Bates, R. A., Kenett, R. S., Steinberg, D. M., and Wynn, H. P. (2006), “Achieving Robust Design from Computer Simulations,” *Quality Technology and Quantitative Management*, 3(2), 161–177.

Bell, J. (2015), “A Simple and Pragmatic Approximation to the Normal Cumulative Probability Distribution,” <http://dx.doi.org/10.2139/ssrn.2579686>, .

Bingham, D., and Sitter, R. R. (2003), “Fractional Factorial Split-Plot Designs for Robust Parameter Experiments,” *Technometrics*, 45(1), 80–89.

Chen, W., Jin, R., and Sudjianto, A. (2006), “Analytical Global Sensitivity Analysis and Uncertainty Propagation for Robust Design,” *Journal of Quality Technology*, 38(4), 333–348.

Del Castillo, E., Alvarez, M. J., Ilzarbe, L., and Viles, E. (2007), “A New Criterion for Robust Parameter Design Experiments,” *Journal of Quality Technology*, 39, 279–295.

Dette, H., and Pepelyshev, A. (2010), “Generalized Latin Hypercube Design for Computer Experiments,” *Technometrics*, 52(4), 421–429.

Fang, K.-T., and Wang, Y. (1994), *Number-Theoretic Methods in Statistics*, London: Chapman and Hall.

Fasshauer, G., and McCourt, M. (2016), *Kernel-based Approximation Methods using MATLAB*, New Jersey, NJ: World Scientific.

Ginsburg, H., and Ben-Gal, I. (2006), “Designing experiments for robust-optimization problems: the V s-optimality criterion,” *IIE Transactions*, 38, 445–461.

Handcock, S. M. (2007), “On cascading latin hypercube designs and additive models for experiments,” *Communications in Statistics - Theory and Methods*, 20, 417–439.

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990), “Minimax and Maximin Distance Designs,” *Journal of Statistical Planning and Inference*, 26(2), 131–148.

Joseph, V. R. (2004), “Quality Loss Functions for Nonnegative Variables and Their Applications,” *Journal of Quality Technology*, 36(2), 129–138.

Joseph, V. R. (2016), “Space-Filling Designs for Computer Experiments: A Review (with discussions),” *Quality Engineering*, 28(1), 28–35.

Joseph, V. R., Gul, E., and Ba, S. (2015), “Maximum Projection Designs for Computer Experiments,” *Biometrika*, 102, 371–380.

Kang, L., and Joseph, V. R. (2009), “Bayesian Optimal Single Arrays for Robust Parameter Design,” *Technometrics*, 51(3), 250–261.

Lehman, J. S., Santner, T. J., and Notz, W. I. (2004), “Designing Computer Experiments to Determine Robust Control Variables,” *Statistica Sinica*, 14, 571–590.

Mak, S., and Joseph, V. R. (2017a), “Minimax and Minimax Projection Designs Using Clustering,” *Journal of Computational and Graphical Statistics*, to appear, DOI:10.1080/10618600.2017.1302881.

Mak, S., and Joseph, V. R. (2017b), “Support Points,” *Annals of Statistics*, to appear, <https://arxiv.org/abs/1609.01811>.

McKay, M., Beckman, R., and Conover, W. (1979), “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” *Technometrics*, 21(2), 239–245.

Morris, M. D., and Mitchell, T. J. (1995), “Exploratory Designs for Computational Experiments,” *Journal of Statistical Planning and Inference*, 43(3), 381–402.

Müller, W. G., Pronzato, L., , and Waldl, H. (2012), “Relations between designs for prediction and estimation in random fields: an illustrative case,” *Advances and Challenges in Space-time modelling of natural events, Lecture notes in Statistics*, 207, 125–139.

Müller, W. G., Pronzato, L., Rendas, J., , and Waldl, H. (2014), “Efficient prediction designs for random fields,” *Applied Stochastic Models in Business and Industry*, 31, 178–194.

Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. (2016), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, New York: Wiley.

Pratola, M., Harari, O., Bingham, D., and Flowers, G. (2016), “Design and Alalysis of Experiments on Non-Convex Regions,” *Technometrics*, <http://dx.doi.org/10.1080/00401706.2016.1164558>.

Sacks, J., Schiller, S. B., and Welch, W. J. (1989), “Designs for Computer Experiments,” *Technometrics*, 31(1), 41–47.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), “Design and Analysis of Computer Experiments,” *Statistical Science*, 4(4), 409–423.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer.

Shoemaker, A. C., Tsui, K.-L., and Wu, C. F. J. (1991), “Economical Experimentation Methods for Robust Design,” *Technometrics*, 33(4), 415–427.

Taguchi, G. (1987), *System of Experimental Design*, White Plains, FL: Unipub/Kraus International.

Tan, M. H. Y. (2015), “Robust Parameter Design With Computer Experiments Using Orthonormal Polynomials,” *Technometrics*, 57(4), 468–478.

Tan, M. H. Y., and Wu, C. F. J. (2012), “Robust Design Optimization with Quadratic Loss Derived from Gaussian Process Models,” *Technometrics*, 54(1), 51–63.

Trefethen, L. N. (2013), *Approximation Theory and Approximation Practice*, Philadelphia, PA: SIAM.

Welch, W., Yu, T., Kang, S., and Sacks, J. (1990), “Computer experiments for quality control by parameter design,” *Journal of Quality Technology*, 22, 15–22.

Williams, B. J., Santner, T. J., and Notz, W. I. (2000), “Sequential Design of Computer Experiments to Minimize Integrated Response Functions,” *Statistica Sinica*, 10(4), 1133–1152.

Wu, C. F. J., and Hamada, M. S. (2009), *Experiments: Planning, Analysis, and Optimization*, New York: Wiley.

Wu, C. F. J., and Zhu, Y. (2003), “Optimal Selection of Single Arrays for Parameter Design Experiments,” *Statistica Sinica*, 13(4), 1179–1199.

Zador, P. L. (1982), “Asymptotic Quantization Error of Continuous Signals and the Quantization Dimension,” *IEEE Transactions on Information Theory*, 28, 139–149.

Zhigljavsky, A., Dette, H., and Pepelyshev, A. (2010), “A New Approach to Optimal Design for Linear Models With Correlated Observations,” *Journal of the American Statistical Association*, 105(491), 1093–1103.