DESIGNS FOR COMPUTER EXPERIMENTS AND UNCERTAINTY QUANTIFICATION

A Thesis
Presented to
The Academic Faculty

by

Evren Gul

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2016

DESIGNS FOR COMPUTER EXPERIMENTS AND UNCERTAINTY QUANTIFICATION

Approved by:

Roshan Joseph Vengazhiyil, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

C. F. Jeff Wu
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Benjamin Haaland
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Shreyes N. Melkote
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

William Brenneman
Statistics Department
Procter & Gamble Company

Date Approved: 29 April 2016

To Serife

ACKNOWLEDGEMENTS

Beautiful people have accompanied me during my doctoral studies. They have made this work possible and this journey pleasant.

First, it is a great pleasure to express my sincere gratitude to my advisor, Professor Roshan Joseph Vengazhiyil for his devoted supervision, insightful guidance and incredible assistance. His kindness, patience and passion for research have made a remarkable impact on my life. It has been a privilege to work with him.

I am also grateful to Professor C. F. Jeff Wu. His guidance and advice are invaluable. He is a great thinker, any talk of him was influential.

I would like to thank Professors Shreyes N. Melkote, Ben Haaland and William Brenneman for their dedication and constructive suggestions on my dissertation. I am thankful to my friend, Dr. Shan Ba, for his great advice and help on my research projects. I am also grateful to Third Wave Systems LLC for the assitance with my research.

I thank to my friends. Your accompany has been very valuable to me in every aspect. I will always remember the great moments we have had.

I would like to express my gratitude to my parents for their endless love and support along the way.

Lastly and most importantly, to Serife, thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Computer experiments are widely-used in analysis of real systems where physical experiments are infeasible or unaffordable. Computer models are usually complex and computationally demanding, consequently, time consuming to run. Therefore, surrogate models, also known as emulators, are fitted to approximate these computationally intensive computer models. Since emulators are easy-to-evaluate they may replace computer models in the actual analysis of the systems. Experimental design for computer simulations and modeling of simulated outputs are two important aspects of building accurate emulators.

This thesis consists of three chapters, covering topics in design of computer experiments and uncertainty quantification of complex computer models. The first chapter proposes a new type of space-filling designs for computer experiments, and the second chapter develops an emulator-based approach for uncertainty quantification of machining processes using their computer simulations. Finally, third chapter extends the experimental designs proposed in the first chapter and enables to generate designs with both quantitative and qualitative factors.

In design of computer experiments, space-filling properties are important. The traditional maximin and minimax distance designs consider only space-fillingness in the full-dimensional space which can result in poor projections onto lower-dimensional spaces, which is undesirable when only a few factors are active. On the other hand, restricting maximin distance design to the class of Latin hypercubes can improve one-dimensional projections but cannot guarantee good space-filling properties in larger subspaces. In the first chapter, we propose designs that maximize space-filling properties on projections to all subsets of factors. Proposed designs are called maximum projection designs. Maximum projection designs have better space-filling properties in their projections compared to other widely-used space-filling designs. They also provide certain advantages in Gaussian process modeling. More importantly, the design criterion can be computed at a cost no more than that of a design criterion which ignores projection properties.

In the second chapter, we develop an uncertainty quantification methodology for machining processes with uncertain input factors. Understanding the uncertainty in a machining process using the simulation outputs is important for careful decision making. However, Monte Carlo-based methods cannot be used for evaluating the uncertainty when the simulations are computationally expensive. An alternative approach is to build an easy-to-evaluate emulator to approximate the computer model and run the Monte Carlo simulations on the emulator. Although this approach is very promising, it becomes inefficient when the computer model is highly nonlinear and the region of interest is large. Most machining simulations are of this kind because the output is affected by a large number of parameters including the workpiece material properties, cutting tool parameters, and process parameters. Building an accurate emulator that works for different kinds of materials, tool designs, tool paths, etc. is not an easy task. We propose a new approach, called in-situ emulator, to overcome this problem. The idea is to build an emulator in a local region defined by the user-specified input uncertainty distribution. We use maximum projection designs and Gaussian process modeling techniques for constructing the in-situ emulator. On two solid end milling processes, we show that the in-situ emulator methodology is efficient and accurate in uncertainty quantification and has apparent advantages over other conventional tools.

Computer experiments with quantitative and qualitative factors are prevalent.

In the third chapter, we extend maximum projection designs so that they can accommodate qualitative factors as well. Proposed designs maintain an economic run size and they are flexible in run size, number of quantitative and qualitative factors and factor levels. Their construction is not restricted to a special design class and does not impose any design configuration. A general construction algorithm, which utilizes orthogonal arrays, is developed. We have shown on several simulations that maximum projection designs with both quantitative and qualitative factors have attractive space-filling properties for all of their projections. Their advantages are also illustrated on optimization of a solid end milling process simulation. Finally, we propose a methodology for sequential construction of maximum projection designs which ensures efficient analysis of systems within financial cost and time constraints. The performance of the sequential construction methodology is demonstrated using the optimization of a solid end milling process.

CHAPTER I

MAXIMUM PROJECTION DESIGNS FOR COMPUTER EXPERIMENTS

1.1 Introduction

Space-filling designs are commonly used in deterministic computer experiments. The *maximin* and *minimax* distance designs (Johnson, Moore and Ylvisaker 1990) are two such designs, can be defined as follows. Suppose we want to construct an n-run design in p factors. Let the design region be the unit hypercube $\mathcal{X}$ and let the design be $\boldsymbol{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, where each design point $\boldsymbol{x}_i \in \mathcal{X} = [0,1]^p$. The maximin distance design tries to spread out the design points in $\mathcal{X}$ by maximizing the minimum distance between any two design points:

$$\max_{\boldsymbol{D}} \min_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}} d(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{1}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. On the other hand, the minimax distance design tries to reduce the gaps in $\mathcal{X}$ by minimizing the maximum distance between any point $\boldsymbol{x} \in \mathcal{X}$ and the design:

$$\min_{\boldsymbol{D}} \max_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}, \boldsymbol{D}), \tag{2}$$

where $d(\boldsymbol{x}, \boldsymbol{D}) = \min_{\boldsymbol{x}_i \in \boldsymbol{D}} d(\boldsymbol{x}, \boldsymbol{x}_i)$.

The maximin distance criterion tends to place a large proportion of points at the corners and boundaries of the hypercube $[0,1]^p$, and thus, unlike Latin hypercube designs (McKay, Beckman and Conover 1979), maximin distance designs do not have good projection properties for each factor (similar problem exists with minimax distance designs). Morris and Mitchell (1995) proposed to overcome this problem by searching for the maximin distance design within the class of Latin hypercube designs. They used the criterion

$$\min_{\boldsymbol{D}} \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right\}^{1/k}, \tag{3}$$

for finding the maximin Latin hypercube design (MmLHD), where $k > 0$ is chosen large enough to achieve maximin distance. Although MmLHDs ensure good space-filling in p dimensions and uniform projections in each dimension, their projection properties in $2, \ldots, p-1$ dimensions are not known. By the effect sparsity principle, only a few factors are expected to be active (e.g., Wu and Hamada 2009). Since the active factors are unknown before the experiment, good projections to all subspaces of the factors are important. However, little research has been done in trying to find space-filling designs that ensure good projections to subspaces of the factors. Tang (1993) and Owen (1994) proposed orthogonal array-based LHDs, which can ensure good t-dimensional projections by using an orthogonal array for strength t. However, the number of runs can explode as t increases and therefore, orthogonal arrays with more than strength two is difficult to use in practice. Moon, Dean and Santner (2011) proposed a design that maximizes the minimum distance in two dimensional projections. Although it has good performance in two- and three- dimensional projections, its performance quickly deteriorates for the larger projection dimensions. This clearly shows that focusing on lower dimensional projections alone is not enough for creating a good design. Draguljic, Santner and Dean (2012) proposed a criterion incorporating projection properties in (3):

$$\min_{\boldsymbol{D}} \left[\frac{1}{\binom{n}{2} \sum_{q \in J} \binom{p}{q}} \sum_{q \in J} \sum_{r=1}^{\binom{p}{q}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left\{ \frac{q^{1/2}}{d_{qr}(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right\}^k \right]^{1/k}, \tag{4}$$

where $d_{qr}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the rth projection of q factors, $q \in J \subseteq \{1, 2, \ldots, p\}$. However, (4) is difficult to compute for large p, so Draguljic, Santner and Dean (2012) had to focus their attention to subspaces with no more than two factors.

A uniformity measure is another type of space-filling criterion (Fang, Li and Sudjianto 2006, ch. 3). The idea is to spread the design points in the space so that the empirical distribution of the points is uniform on $[0,1]^p$. Hickernell (1998) proposed the centered L_2-discrepancy (CL_2) criterion, which ensures good projections to all subspaces. It can be computed using the following simplified formula:

$$\begin{aligned}\min_{\boldsymbol{D}} \quad & \left(\frac{13}{12}\right)^p - \frac{2}{n}\sum_{l=1}^{n}\prod_{j=1}^{p}\left(1+\frac{1}{2}|x_{lj}-.5|-\frac{1}{2}|x_{lj}-.5|^2\right) \\ & +\frac{1}{n^2}\sum_{l=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{p}\left(1+\frac{1}{2}|x_{li}-.5|+\frac{1}{2}|x_{ji}-.5|-\frac{1}{2}|x_{li}-x_{ji}|\right). \qquad (5)\end{aligned}$$

Although uniform designs (UDs) are useful for approximating integrals, it is not clear if they are as good as MmLHDs for approximating functions. In fact, Dette and Pepelyshev (2010) have shown that placing more points in the boundaries than around the center can minimize the prediction errors from Gaussian process (GP) modeling. They proposed a logarithmic potential energy criterion for finding optimal LHDs, which is similar in spirit to the MmLHDs. In fact, their criterion can be obtained as the limit of the criterion in (3) with $k \to 0$.

In this chapter we propose a new space-filling criterion that incorporates projection properties to all subspaces of factors. Our criterion is closely related to (4), but much simpler to compute. In fact, our criterion can be computed at a cost no more than that of (3) and hence we are able to efficiently construct our proposed designs.

The chapter is organized as follows. In Section 1.2, we develop our design criterion and in Section 1.3, we discuss a computational algorithm for finding the optimal designs. Some numerical comparisons using distance and uniformity measures are done with MmLHDs, generalized maximin Latin hypercube designs (GMmLHDs) (Dette and Peppelyshev 2010) and UDs in Section 1.4. In Section 1.5, we investigate the performance of the proposed designs in GP modeling and we conclude with some remarks in Section 1.6.

1.2 Maximum Projection Designs

When a design is projected onto a subspace, the distances between the points are calculated with respect to the factors that define the subspace. Therefore, define a weighted Euclidean distance between the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) = \left\{ \sum_{l=1}^{p} \theta_l (x_{il} - x_{jl})^2 \right\}^{1/2},$$

where $\theta_l = 1$ for the factors defining the subspace and $\theta_l = 0$ for the remaining factors. It makes sense to use weights between 0 and 1, which can be viewed as measures of importance for the factors. Let $0 \leq \theta_l \leq 1$ be the weight assigned to factor l and let $\sum_{l=1}^{p} \theta_l = 1$. Then, the criterion in (3) can be modified to

$$\min_{\boldsymbol{D}} \phi_k(\boldsymbol{D}; \boldsymbol{\theta}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})}, \tag{6}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{p-1})'$ and $\theta_p = 1 - \sum_{l=1}^{p-1} \theta_l$. We omit the power $1/k$ in (3) because we are interested only in finite values of k. Unfortunately, we have no idea about the importance of the factors before the experiment, so there is no easy way to choose $\boldsymbol{\theta}$. One approach to overcome this is to adopt a Bayesian framework, i.e., to assign a prior distribution for $\boldsymbol{\theta}$ and then to minimize the expected value of the objective function.

Assuming equal importance to all factors a priori, we take the prior distribution for $\boldsymbol{\theta}$ to be

$$p(\boldsymbol{\theta}) = \frac{1}{(p-1)!}, \boldsymbol{\theta} \in S_{p-1}, \tag{7}$$

where $S_{p-1} = \{\boldsymbol{\theta} : \theta_1, \ldots, \theta_{p-1} \geq 0, \sum_{i=1}^{p-1} \theta_i \leq 1\}$. Thus, our design criterion becomes

$$\min_{\boldsymbol{D}} E\{\phi_k(\boldsymbol{D}; \boldsymbol{\theta})\} = \int_{S_{p-1}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})} p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{8}$$

In general, this is not easy to evaluate. However, we can perform the integration analytically for a special case of k as shown below. All the proofs are given in the Appendix.

Theorem 1. *If $k = 2p$, then under the prior in (7)*

$$E\{\phi_k(\boldsymbol{D};\boldsymbol{\theta})\} = \frac{1}{\{(p-1)!\}^2}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{\prod_{l=1}^{p}(x_{il}-x_{jl})^2}. \tag{9}$$

Thus, we propose a new criterion:

$$\min_{\boldsymbol{D}} \psi(\boldsymbol{D}) = \left\{\frac{1}{\binom{n}{2}}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{\prod_{l=1}^{p}(x_{il}-x_{jl})^2}\right\}^{1/p}. \tag{10}$$

For any l, if $x_{il} = x_{jl}$ for $i \neq j$, then $\psi(\boldsymbol{D}) = \infty$. Therefore, the design that minimizes $\psi(\boldsymbol{D})$ must have n distinct levels for each factor. Furthermore, because the denominator of (34) has products of squared distances from all the factors, no two points can get close to each other in any of the projections. Thus, the design that minimizes $\psi(\boldsymbol{D})$ tends to maximize its projection capability in all subspaces of factors. Therefore, we call the optimal design a maximum projection (MaxPro) design.

We need to make some comments about the choice of k in Theorem 1. Audze and Eglais (1977) proposed a criterion similar to (3) with $k = 2$, which is motivated by an analogy to minimizing the total force among electrically charged particles. On the other hand, Morris and Mitchell (1995) suggested to use small but large enough value of k in the range of 1 to 100 to obtain maximin distance designs with minimum index. The simulations (for low dimensional cases) performed in Joseph et al. (2014) show that $k = 4p$ gives approximately uniform designs. Thus, the choice $k = 2p$ used in the Theorem 1 is quite reasonable and is not just made for mathematical convenience.

Before discussing the computational algorithm for finding the optimal MaxPro designs, we would like to mention about an extension of the proposed criterion. Consider a generalized weighted distance measure:

$$d(\boldsymbol{x}_i,\boldsymbol{x}_j;\boldsymbol{\theta}) = \left\{\sum_{l=1}^{p}\theta_l|x_{il}-x_{jl}|^s\right\}^{\frac{1}{s}},$$

where $s = 2$ gives the previous weighted Euclidean distance. Now proceeding as before, we obtain a new criterion:

$$\min_{\boldsymbol{D}} \psi_s(\boldsymbol{D}) = \left\{\frac{1}{\binom{n}{2}}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{\prod_{l=1}^{p}|x_{il}-x_{jl}|^s}\right\}^{\frac{2}{ps}}.$$

As $s \to \infty$, this criterion becomes

$$\max_{\boldsymbol{D}} \min_{i,j} \prod_{l=1}^{p} |x_{il} - x_{jl}|.$$

The similarities of the foregoing two criteria to those in (1) and (3) are very interesting. However, in this paper, we prefer to use the criterion in (34) because we found $\psi_s(\boldsymbol{D})$ to be numerically unstable for large values of s. Moreover, the choice $s = 2$ has some optimality properties for using in Gaussian process modeling, which will be explained in section 1.5.

1.3 Optimal Design Construction Algorithm

Although $\psi(\boldsymbol{D})$ in (34) is easy to compute, finding the MaxPro design by minimizing it is not an easy problem. First, the number of variables in the optimization, np, is extremely large even for moderate-sized problems. Second, the objective function has many local minima because it becomes infinite whenever $x_{il} = x_{jl}$ for any $l = 1, \ldots, p$ and $i \neq j$. Moreover, the design remains the same under the reordering of rows or columns, which produces many local minima. Thus, optimization of (34) directly using a continuous optimization algorithm can easily terminated at a local optimum.

Because a MaxPro design will have n distinct levels for each factor, it can be viewed as a Latin hypercube design, but not necessarily with equally spaced levels. Therefore, we can make use of the algorithms for the construction of optimal Latin hypercube designs such as simulated annealing (Morris and Mitchell 1995). The simulated annealing algorithm starts with an initial LHD, and tries to improve the MaxPro criterion in (34) iteratively. In each iteration a new design, $\boldsymbol{D}_{new}$, is obtained by swapping two randomly chosen elements in a randomly chosen column of $\boldsymbol{D}$ and it replaces $\boldsymbol{D}$ if it brings improvement. Otherwise, it replaces $\boldsymbol{D}$ with probability $\pi = \exp -[\psi(\boldsymbol{D}_{new}) - \psi(\boldsymbol{D})]/t$, where t corresponds to "temperature" which is a user-defined parameter. Because a short-cut formula can be developed for efficiently computing the objective function after the swapping of two elements, this algorithm can be efficiently implemented.

After obtaining the optimal maximum projection Latin hypercube design (MaxProLHD), we apply a continuous optimization algorithm to find the locally optimal MaxPro design in the neighborhood of the optimal MaxProLHD. For doing this, first we use a logit transformation to map the variable in $[0, 1]$ to the real line: $z_{rs} = \log\{x_{rs}/(1-x_{rs})\}$. This allows us to remove the box constraints on the variables. The gradient of the objective function under this transformation can be obtained as

$$\frac{\partial \psi^p(\boldsymbol{D})}{\partial z_{rs}} = \frac{2}{\binom{n}{2}} \sum_{i \neq r} \frac{1}{\prod_{l=1}^{p} (x_{il} - x_{rl})^2} \frac{x_{rs}(1 - x_{rs})}{(x_{is} - x_{rs})},$$

which can be used to implement a fast derivative-based optimization algorithm. An interesting point to note is that such an algorithm cannot be applied for finding MmLHD because of the Latin hypercube constraints. We are able to use an unconstrained optimization algorithm because our objective function automatically satisfies the Latin hypercube constraints. This can be advantageous in certain applications.

As an example, we constructed a MaxPro design with $n = 30$ and $p = 3$. Their two-dimensional projections are shown in Figure 4 along with those of an MmLHD and UD. We can see that the MaxPro designs are visually very appealing. Its design points are spread out evenly in every two-dimensional projections. On the other hand, large blank areas and close points are observed in both MmLHD and UD, which are undesirable. More careful comparisons are made in the next section.

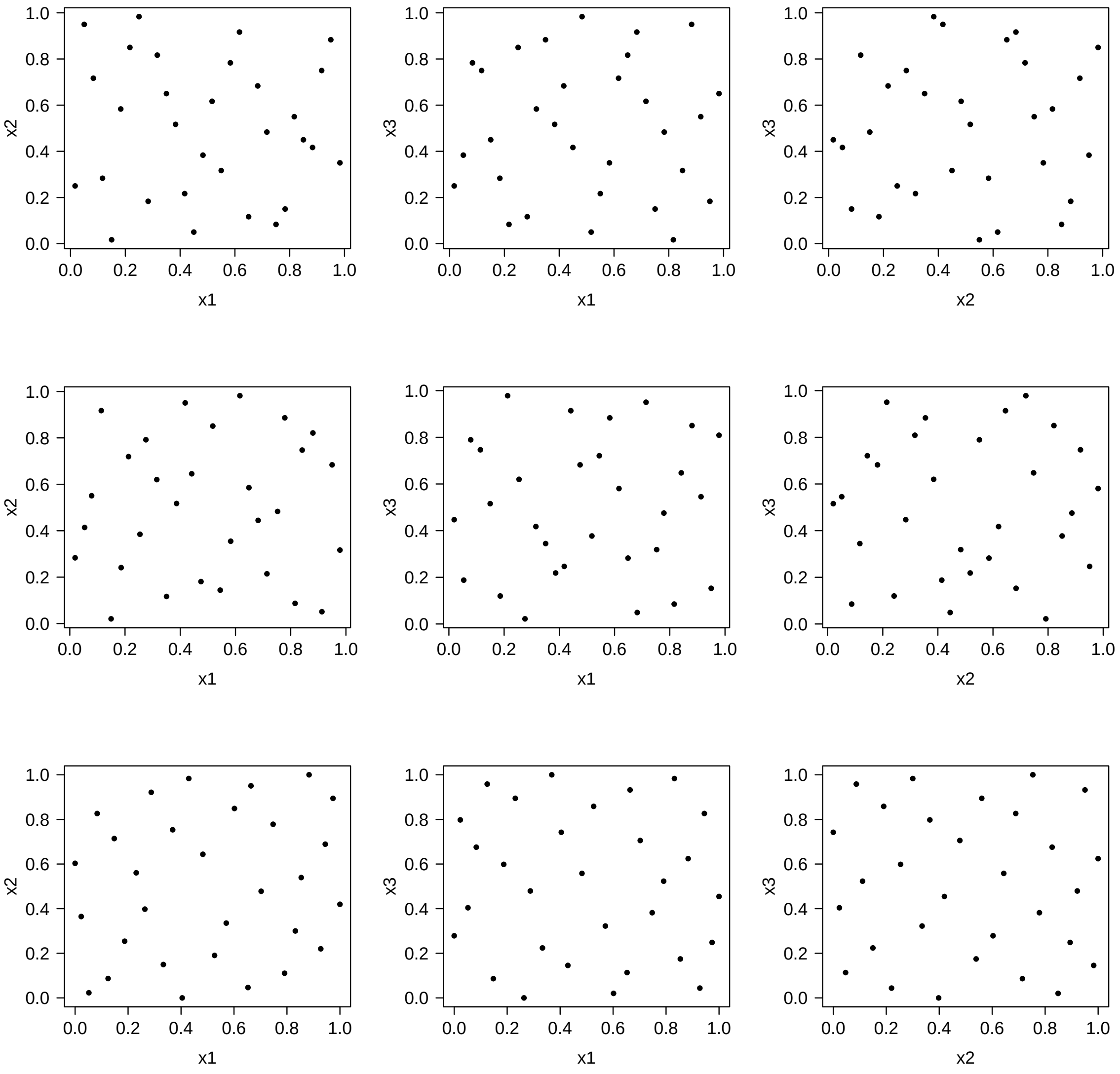

Figure 1: Two-dimensional projections of 30-run 3-factor MmLHD, UD, and MaxPro design, respectively

1.4 Numerical Results

In this section, we compare the performances of MaxPro designs and MaxProLHDs and other popular space-filling designs such as MmLHDs, UDs, and generalized maximin Latin hypercube designs (GMmLHDs) (Dette and Pepelyshev 2010), on nine simulation settings with $p = 5, 10, 20$ and $n = 5p, 10p, 15pand20p$. Because the conclusions from all the nine cases are similar, we report results only for the 100-run 10-factor design. The simulation results for the cases $p = 10$ and $n = 5p$ and

$n = 15p$ are given in Appendix in sections 1.7.2 and 1.7.3, respectively. The MmLHD is constructed using the R package SLHD (Ba 2015), the UD is constructed using the software JMP, and the GMmLHD is constructed using the arc-sine transformation of the MmLHD (Dette and Pepelyshev 2010).

Designs are compared on three measures: minimum distance among the design points, maximum distance from any point in the design space to the nearest design point, and CL_2. These three criteria are summarized for each sub-dimensional projection. For a given sub-dimension q, a measure is computed for $\binom{p}{q}$ possible projections and the worst case is used for comparison.

The minimum of minimum distances among the projections are shown in Figure 20. As expected, MmLHD and GMmLHD have the largest minimum distances in the full design space. On the other hand, the MaxPro design outperforms others when the projection dimension is reduced by one except for one- and two-dimensional projections for MmLHD and one-dimensional projection for GMmLHD. Compared to the MmLHD and GMmLHD, the minimum distances are larger by a factor of 1.48 to 3.43 for dimensions 3-9 and by a factor of 1.20 to 3.09 for dimensions 2-9, respectively, which is a significant improvement. The one- and two-dimensional minimum distances are slightly smaller for MaxPro design because the factor levels are not equally-spaced and are close to each other near the boundaries. This problem is avoided for MaxProLHDs. They have smaller minimum distances compared to the MaxPro design in dimensions 3-9, but they are uniformly better than MmLHD and GMmLHD except for the full dimension. Since CL_2 criterion accounts for projections, UD exhibits better performance than MmLHD and GMmLHD, but still much worse than the MaxPro design and MaxProLHD. The minimum distance measure does not account for the index of the design, i.e., the number of pairs with the minimum distance. A better maximin measure in projection q that also incorporates the maximin index of the design is

$$Mm_q = \min_{r=1,\dots,\binom{p}{q}} \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d_{qr}^{2q}(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right\}^{-1/(2q)},$$

where $d_{qr}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the rth

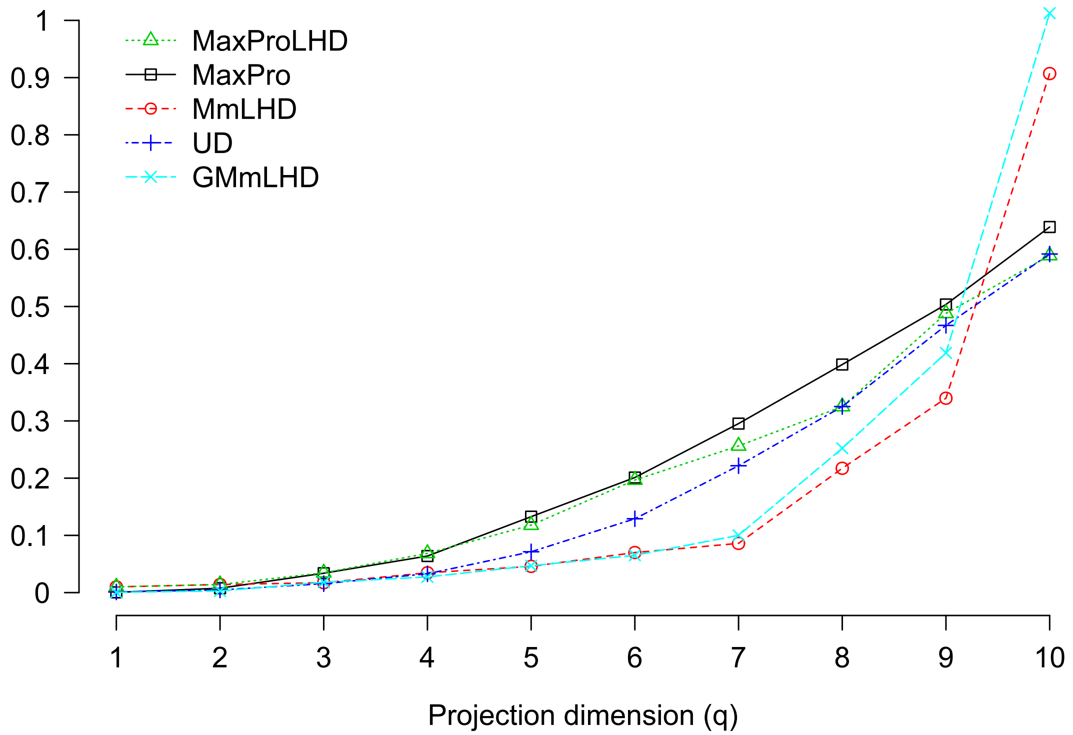

Figure 2: Minimum of minimum distances (larger-the-better)

projection of dimension q. This measure is plotted in Figure 21. Same observations are valid for Mm_q measure. In other words, although the MmLHD and GMmLHDs have the largest Mm_q values in the full design space, the MaxPro design and MaxProLHD has significantly better performance for the lower dimensional projections.

Now consider the maximum distance measure. Its computation is cumbersome because we need to search over the whole space $[0,1]^q$ to find the point having maximum distance to the nearest point in the design. For doing this, we sample N_q points from $[0,1]^q$ and approximate the minimax measure by the observed maximum distance between the points and the design. We use a union of 3^q factorial design with levels $\{0, 0.5, 1\}$ and 2^{16} run Sobol sequence for the sample. Thus, in the full dimensional space, $N_p = 3^{10} + 2^{16} = 124, 585$. The maximum of maximum distance among the projections are shown in Figure 22. There is no clear winner in this case as

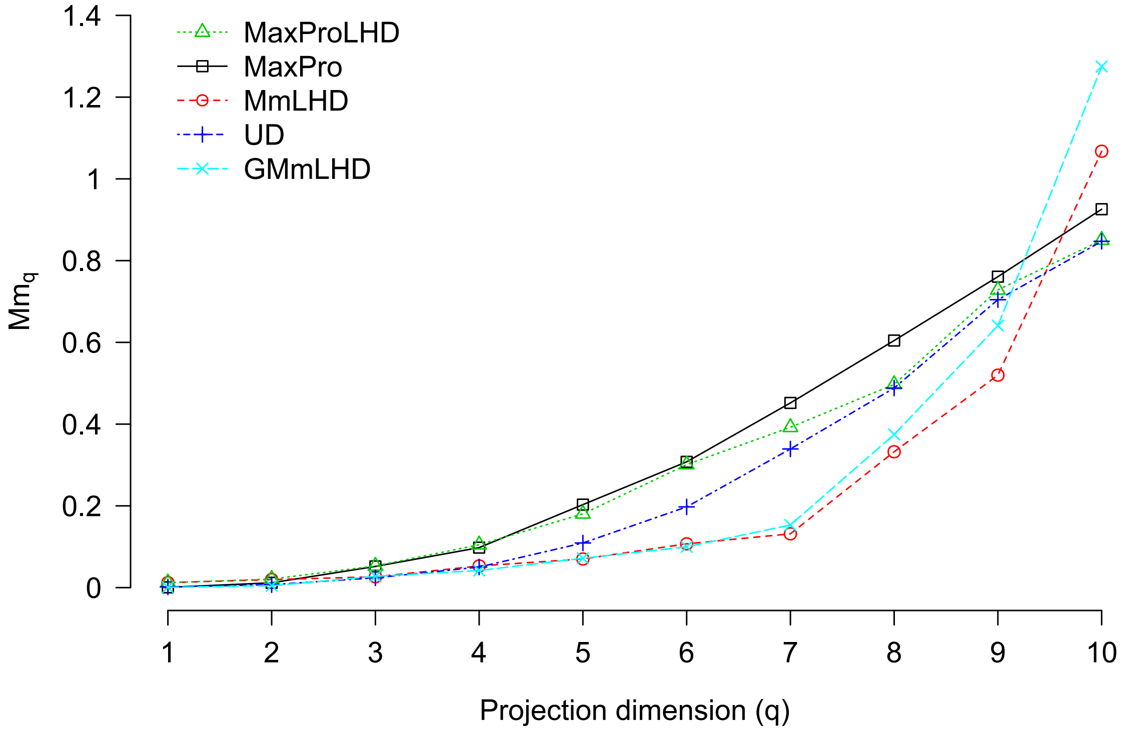

Figure 3: Mm_q (larger-the-better)

the differences among the four designs on this measure are minor. A better minimax measure that also incorporates the minimax index of the design is

$$mM_q = \max_{r=1,\ldots,\binom{p}{q}} \max_{u \in \mathcal{X}_q} \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{d_{qr}^{2q}(u, x_i)} \right\}^{-1/(2q)},$$

where $\mathcal{X}_q$ is the set of sample points with size $N_q = 3^q + 2^{16}$. As can be seen in Figure 23, the conclusions from this measure remains the same as before. Overall, the MaxPro design and MaxProLHD are slightly better in lower dimensional projections and slightly worse in higher dimensional projections compared to MmLHD and UD.

Finally, consider the uniformity measure, CL_2, defined in 5. The maximum CL_2 values among the q-dimensional projections are shown in Figure 24. As expected, the UD performs best under this criterion because it is obtained by minimizing CL_2. The performance of the MaxPro design is significantly worse for the CL_2 criterion

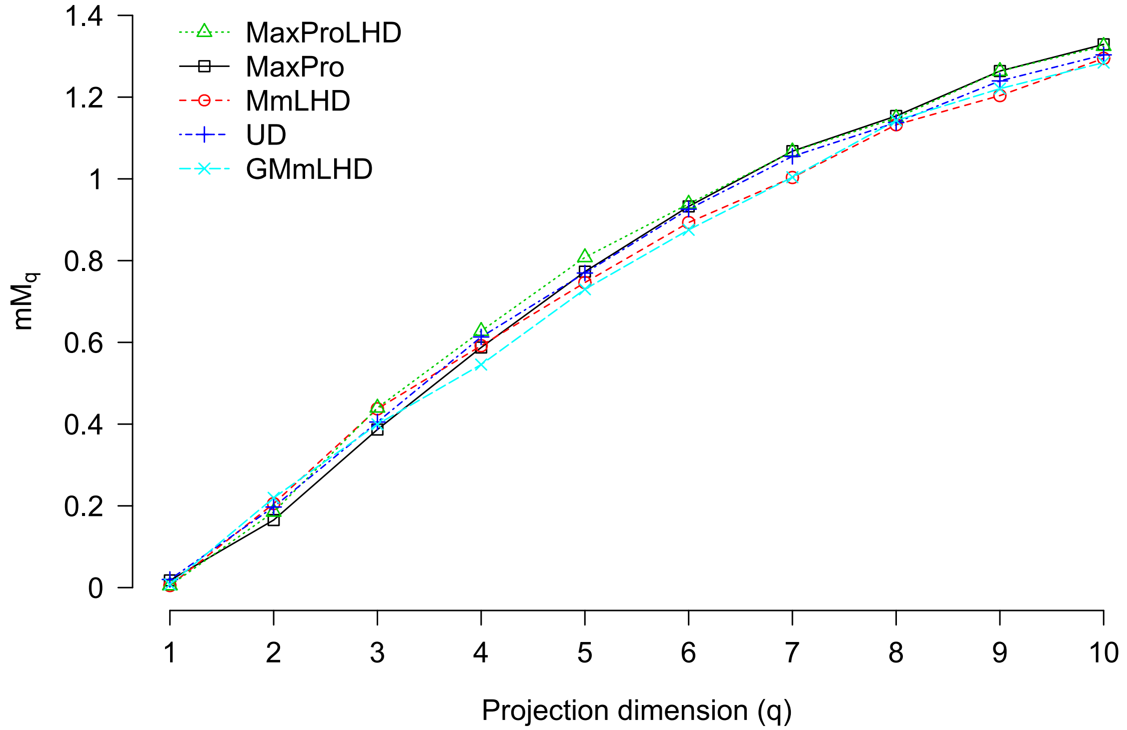

Figure 4: Maximum of maximum distances (smaller-the-better)

compared to UD, but much better than that of the GMmLHD. This is because both MaxPro design and GMmLHDs favor more points towards the boundaries than at the center, which affects the overall uniformity of the points in the design space. Interestingly, the MaxProLHD has much better uniformity than the MaxPro design, over all dimensions. This is surprising because the Latin hypercube restriction only makes the spacing of the levels equal, but somehow it improves the uniformity in all subspaces. We believe that uniformity is not as important as the maximin and minimax distance measures in a computer experiment. This is because the primary objective of a computer experiment is in approximating a computationally expensive computer model. Uniformity becomes important only when we want to approximate an integral using sample averages, which is rarely the objective of a computer experiment. In fact, in many cases, the integrals computed using the approximated model

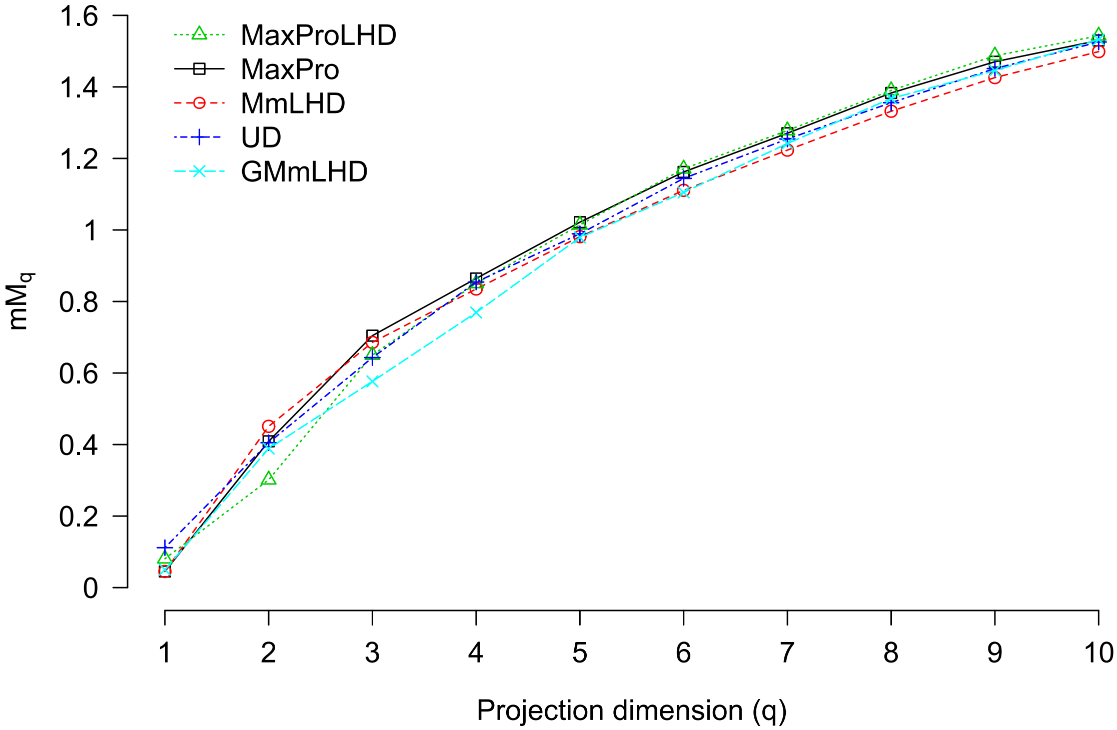

Figure 5: mM_q (smaller-the-better)

can outperform the sample averages. Thus, the poor performance of MaxPro designs under the uniformity measure is not of a great concern.

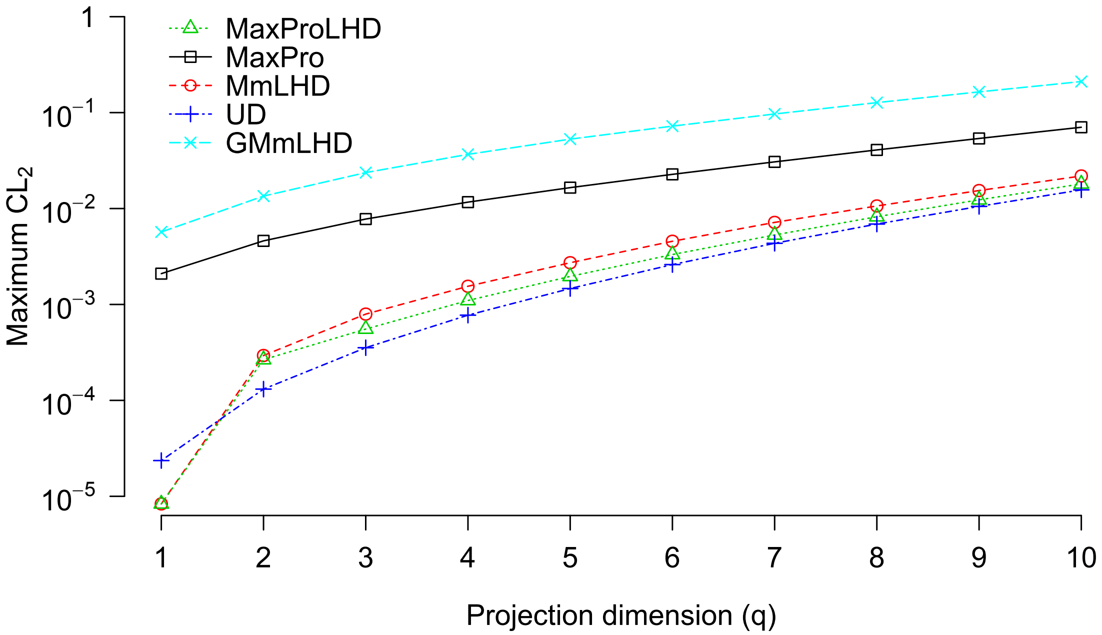

Figure 6: Maximum of Centered-L_2-Discrepancies (smaller-the-better)

1.5 Gaussian Process Modeling

The space-filling designs discussed earlier are model-independent, which allows the experimenter to fit a wide variety of models to the data. On the other hand, better designs can be developed for a specified model class. In computer experiments, Gaussian process modeling or kriging is widely used for approximating the response surface (Sacks, Welch, Mitchell and Wynn 1989; Currin, Mitchell, Morris and Ylvisaker 1991). The ordinary kriging model is

$$Y(\boldsymbol{x}) = \mu + Z(\boldsymbol{x}), \tag{11}$$

where $\boldsymbol{x} \in \mathbb{R}^p$, μ is the overall mean, and $Z(\boldsymbol{x})$ is a stationary Gaussian process with mean zero and covariance function $\sigma^2 R(\cdot)$. A popular choice for $R(\cdot)$ is the Gaussian correlation function,

$$R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\alpha}) = \exp\left\{ -\sum_{l=1}^{p} \alpha_l (x_{il} - x_{jl})^2 \right\}, \alpha_l \in (0, \infty), l = 1, \ldots, p. \tag{12}$$

A popular model-based design is the maximum entropy design (MED) (Shewry and Wynn 1987) is given by

$$\max_{\boldsymbol{D}} det(\boldsymbol{R}(\boldsymbol{\alpha})), \tag{13}$$

where $\boldsymbol{R}(\boldsymbol{\alpha})$ is the correlation matrix with ijth element $R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\alpha})$. A major drawback with such a model-based design is that we need to specify the value of $\boldsymbol{\alpha}$ for finding the optimal design, which is unknown before conducting the experiment. One can use a guessed value of $\boldsymbol{\alpha}$ for finding the optimal design, but the designs can be very poor if the guess is badly wrong. One way to mitigate this problem is to assign a prior distribution for $\boldsymbol{\alpha}$ and optimize the expected value of the objective function. Let us assume a noninformative prior for $\boldsymbol{\alpha}$:

$$p(\boldsymbol{\alpha}) \propto 1, \boldsymbol{\alpha} \in \mathbb{R}_+^p. \tag{14}$$

The following is an optimality result for MaxPro designs relating to GP modeling.

Theorem 2. *For the Gaussian correlation function and the noninformative prior for $\boldsymbol{\alpha}$ in (14), MaxPro designs minimize*

$$E\{\sum_{i=1}^{n} \sum_{j \neq i} \boldsymbol{R}_{ij}^{\gamma}(\boldsymbol{\alpha})\}$$

for any $\gamma > 0$.

In other words, MaxPro designs minimize the expected sum of off-diagonal elements (or any power of them) of the correlation matrix. Although this result is not directly related to the maximum entropy criterion, there seems to be some connections. An application of Hadamard's inequality and Gershgorin's theorem gives the following bounds on the $det(\boldsymbol{R}(\boldsymbol{\alpha}))$

$$\prod_{s=1}^{n} (1 - \sum_{j \neq i_s} \boldsymbol{R}_{i_s j}(\boldsymbol{\alpha})) \leq det(\boldsymbol{R}(\boldsymbol{\alpha})) \leq 1,$$

where $i_1, \ldots, i_n \in \{1, 2, \ldots, n\}$. Thus, minimizing the off-diagonal elements of the $\boldsymbol{R}(\boldsymbol{\alpha})$ matrix tends to increase the lower bound on the $det(\boldsymbol{R}(\boldsymbol{\alpha}))$, and the upper bound is achieved when all of the off-diagonal elements are 0. Therefore, the MaxPro designs are expected to perform well on the maximum entropy criterion. The nice thing is that we do not need to specify any value for the correlation parameters to obtain MaxPro designs.

To compare the performance of MaxPro design with MED, we generated a 100-run 10-factor MED using the software JMP, where the correlation parameter α_l is set to be equal to 5 for $l = 1, \ldots, 10$. Now we compute the minimum determinant among the q-dimensional projections, $\min_r \log det(\boldsymbol{R}^{q,r}(\boldsymbol{\alpha}))$, where $\boldsymbol{R}^{q,r}$ is the correlation matrix calculated for the rth q-dimensional projection. As seen in Figures 25 and 26, the MaxPro design is better than the MED in lower dimensional projections and has comparable performance in higher dimensional projections. MaxProLHD is worse than MaxPro design but is uniformly better than MmLHD.

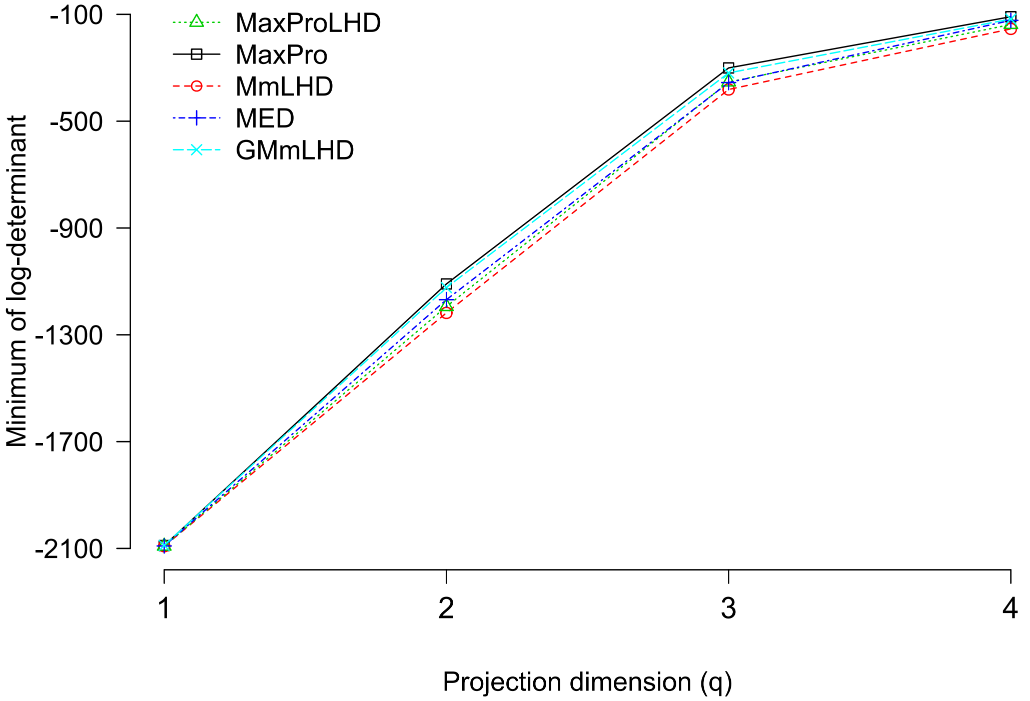

Figure 7: Minimum of log-determinant for projection dimensions 1 to 4 (larger-the-better)

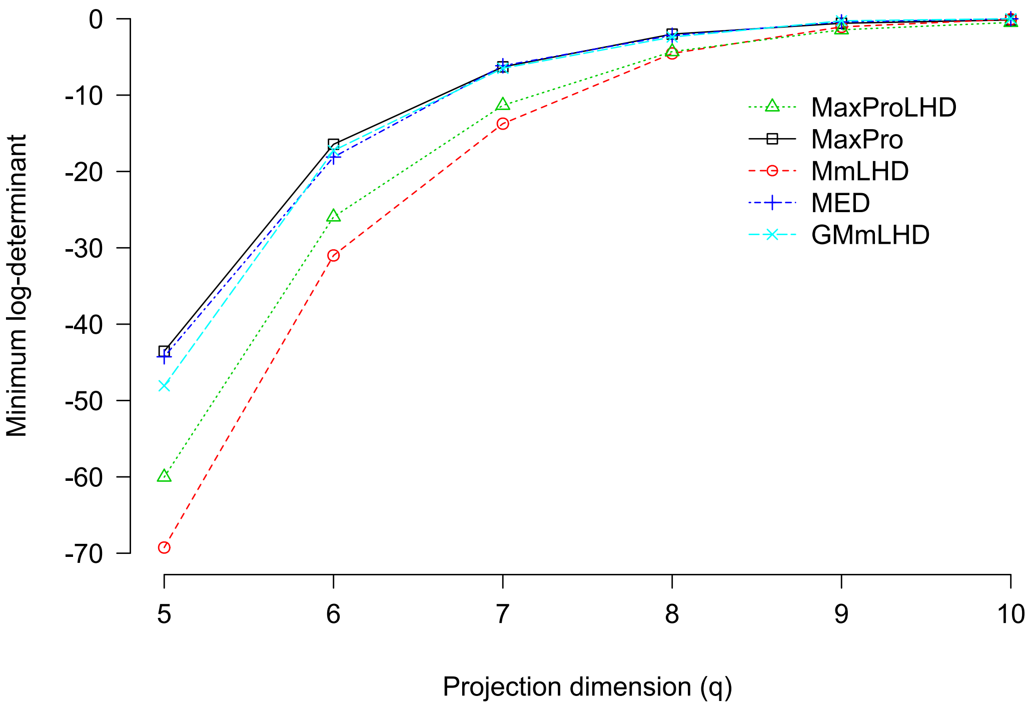

Figure 8: Minimum of log-determinant for projection dimensions 5 to 10 (larger-the-better)

The ordinary kriging predictor is

$$\hat{y}(\boldsymbol{x}) = \hat{\mu}(\boldsymbol{\alpha}) + \boldsymbol{r}(\boldsymbol{x};\boldsymbol{\alpha})'\boldsymbol{R}^{-1}(\boldsymbol{\alpha})\{y - \hat{\mu}(\boldsymbol{\alpha})\mathbf{1}\},$$

where $\hat{\mu}(\boldsymbol{\alpha}) = \mathbf{1}'\boldsymbol{R}^{-1}(\boldsymbol{\alpha})\boldsymbol{y}/\mathbf{1}'\boldsymbol{R}^{-1}(\boldsymbol{\alpha})\mathbf{1}$, $\boldsymbol{r}(\boldsymbol{x};\boldsymbol{\alpha})$ is an $n \times 1$ vector with ith element $R(\boldsymbol{x}-\boldsymbol{x}_i;\boldsymbol{\alpha})$, $\boldsymbol{y} = (y_1, \ldots, y_n)'$ is the experimental data, and $\mathbf{1}$ is a vector of 1's having length n. We can see that the predictor involves the inverse of the correlation matrix at the optimal value of $\boldsymbol{\alpha}$. The optimal value of $\boldsymbol{\alpha}$ is estimated using likelihood or cross validation based methods. For example, the maximum likelihood estimate can be obtained by minimizing negative log-likelihood (Santner, Williams and Notz 2003, p. 66)

$$\log det \boldsymbol{R}(\boldsymbol{\alpha}) + n \log \hat{\sigma}^2(\boldsymbol{\alpha})$$

with respect to $\boldsymbol{\alpha}$, where $\hat{\sigma}^2(\boldsymbol{\alpha}) = \{\boldsymbol{y} - \hat{\mu}(\boldsymbol{\alpha})\mathbf{1}\}'\boldsymbol{R}^{-1}(\boldsymbol{\alpha})\{y - \hat{\mu}(\boldsymbol{\alpha})\mathbf{1}\}/n$. This again requires the computation of $\boldsymbol{R}^{-1}(\boldsymbol{\alpha})$, but now for many values of $\boldsymbol{\alpha}$. The matrix inverse operation can become difficult and unstable at some values of $\boldsymbol{\alpha}$, so it will be good to use an experimental design that avoids this for all values of $\boldsymbol{\alpha}$. The instability of a matrix inverse can be assessed using its condition number. Figure 27 shows the maximum of condition numbers among the q-dimensional projections after adding a small nugget, 10^{-6}, to the diagonals of $\boldsymbol{R}(\boldsymbol{\alpha})$, where $\alpha_l = 5$ for $l = 1, \ldots, 10$. We can see that MaxPro design has superior performance over the other designs.

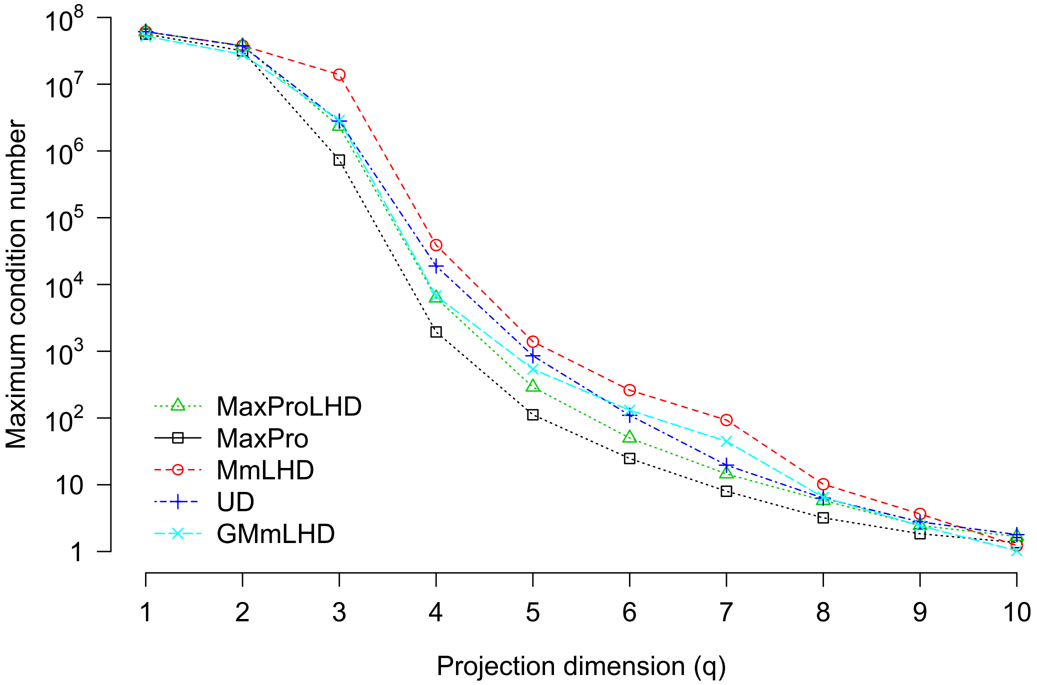

Figure 9: Maximum of condition numbers of $\boldsymbol{R}(\boldsymbol{\alpha})$ (smaller-the-better)

The maximum prediction variance in the design space can be used as another criterion for evaluating a design. For ordinary kriging, the prediction variance is proportional to

$$1 - \boldsymbol{r}(\boldsymbol{x};\boldsymbol{\alpha})'\boldsymbol{R}^{-1}(\boldsymbol{\alpha})\boldsymbol{r}(\boldsymbol{x};\boldsymbol{\alpha}) + \frac{\{1 - \boldsymbol{r}(\boldsymbol{x};\boldsymbol{\alpha})'\boldsymbol{R}^{-1}(\boldsymbol{\alpha})\mathbf{1}\}^2}{\mathbf{1}'\boldsymbol{R}^{-1}(\boldsymbol{\alpha})\mathbf{1}}.$$

The maximum prediction variance among the q-dimensional projections can now be approximated using the same set of N_q points used earlier in approximating the minimax measure and is plotted in Figure 28 for projection dimensions 1 to 6. The MaxPro design and the GMmLHD are the two winners on this performance measure.

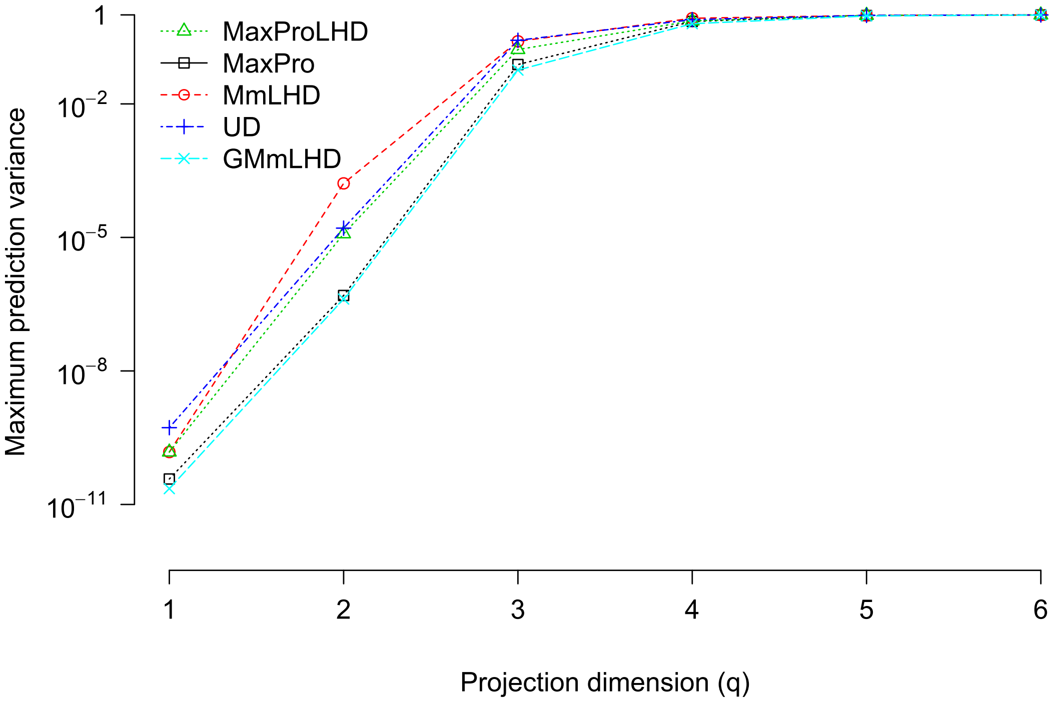

Figure 10: Maximum of prediction variance of ordinary kriging (smaller-the-better)

1.6 Conclusions

In this work we have proposed a new experimental design for computer experiments, namely MaxPro design, which ensures good projection properties in all subspaces of the factors. Through several simulations and various types of measures we have shown that they perform better or at least as good as the popular space-filling designs such as maximin Latin hypercube design (MmLHD) and uniform design (UD). Their advantages for use in Gaussian process modeling are also established.

In our numerical studies, we found that MaxProLHD has close performance to MaxPro design. In fact, it has surprisingly better performance under uniformity measures, although uniformity may not be of great importance in many applications. However, due to its simpler construction and better one-dimensional projections, MaxProLHD may be a more attractive choice than MaxPro design in most applications.

1.7 *Appendix*

1.7.1 Proofs of Theorems 1 and 2

Proof of Theorem 1

Proof. For $k = 2p$, we have

$$E\{\phi_k(\boldsymbol{D};\boldsymbol{\theta})\} = \frac{1}{(p-1)!}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\int_{S_{p-1}}\left\{\sum_{l=1}^{p}\theta_l(x_{il}-x_{jl})^2\right\}^{-p} d\boldsymbol{\theta}.$$

Let

$$Q_p(p,a) = \int_{S_{p-1}}\left\{\sum_{l=1}^{p-1}\theta_l d_l + (1-\sum_{l=1}^{p-1}\theta_l)a\right\}^{-p} d\boldsymbol{\theta}.$$

For $a \neq d_{p-1}$, integrating with respect to $\boldsymbol{\theta}_{p-1}$,

$$Q_p(p,a) = \frac{1}{(p-1)(a-d_{p-1})}\left\{Q_{p-1}(p-1,d_{p-1}) - Q_{p-1}(p-1,a)\right\}. \tag{15}$$

Assume

$$Q_{p-1}(p-1,a) = \frac{1}{(p-2)!d_1\ldots d_{p-2}a}. \tag{16}$$

Then, from (15), $Q_p(p,a) = 1/\{(p-1)!d_1\ldots d_{p-2}d_{p-1}a\}$. This result holds for all a including $a = d_{p-1}$. Since $Q_2(2,a) = 1/(d_1 a)$, by mathematical induction, (16) is true for all p. Now the result follows because $Q_p(p,d_p) = 1/\{(p-1)!d_1\ldots d_p\}$.

$\diamondsuit$

Proof of Theorem 2

Proof. For any $\gamma > 0$, we have

$$\begin{aligned} E(\sum_{i=1}^{n}\sum_{j\neq i}\boldsymbol{R}_{ij}^{\gamma}) &= \sum_{i=1}^{n}\sum_{j\neq i} E\left\{\prod_{l=1}^{p} e^{-\gamma\alpha_l(x_{il}-x_{jl})^2}\right\} \\ &= \sum_{i=1}^{n}\sum_{j\neq i}\left\{\prod_{l=1}^{p}\int_0^{\infty} e^{-\gamma\alpha_l(x_{il}-x_{jl})^2} d\alpha_l\right\} \\ &= \frac{1}{\gamma^p}\sum_{i=1}^{n}\sum_{j\neq i}\frac{1}{\prod_{l=1}^{p}(x_{il}-x_{jl})^2}, \end{aligned}$$

which is minimized by a maximum projection design. ◇

1.7.2 Numerical results for case 2: p=10 and n=5p=50

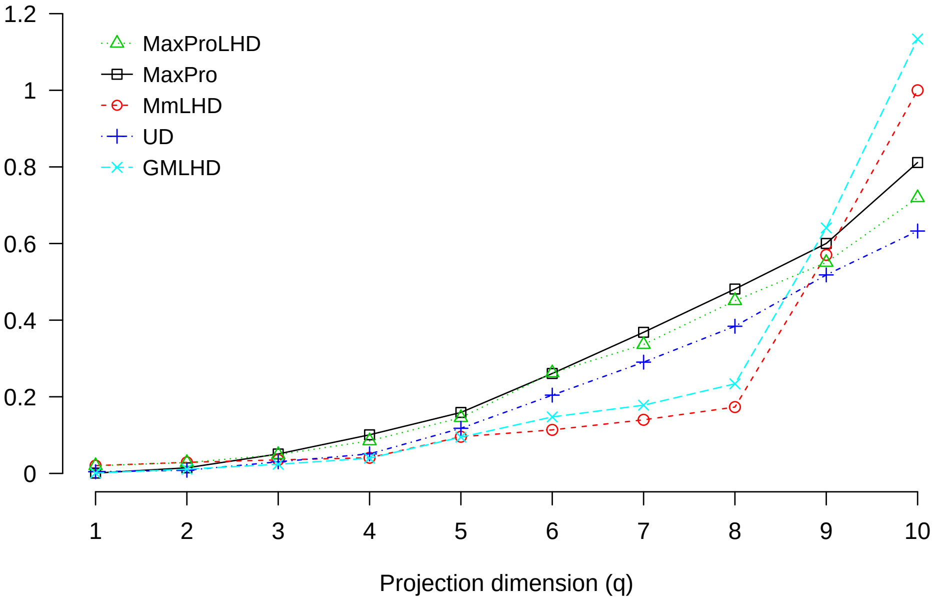

Figure 11: Case 2: Minimum of minimum distances (larger-the-better)

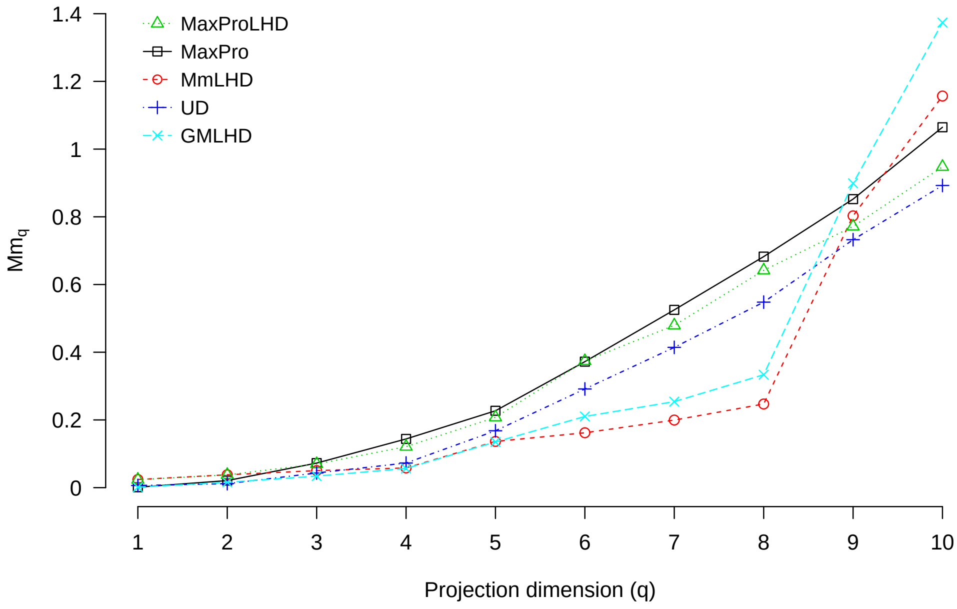

Figure 12: Case 2: Mm_q (larger-the-better)

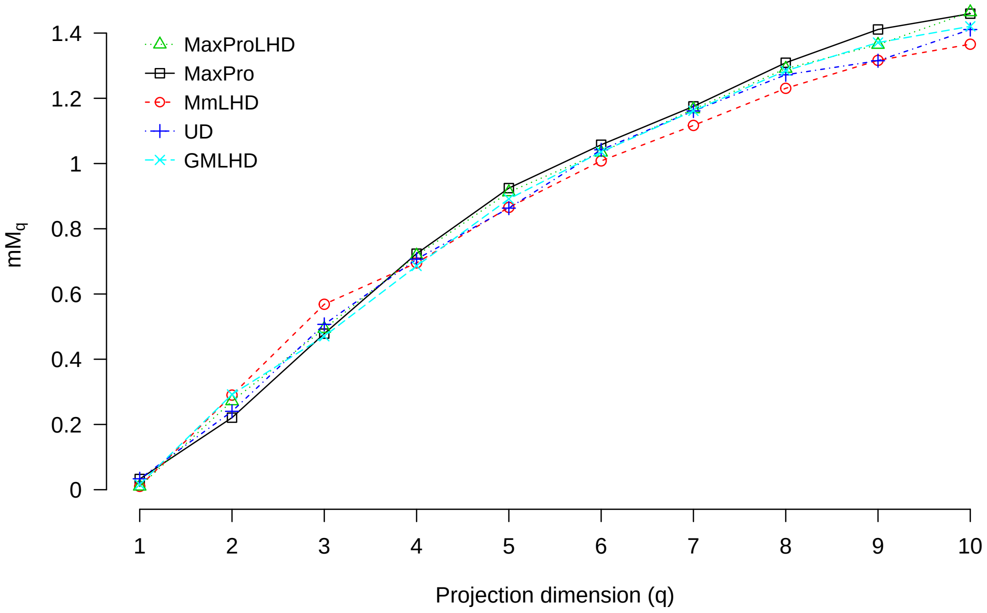

Figure 13: Case 2: Maximum of maximum distances (smaller-the-better)

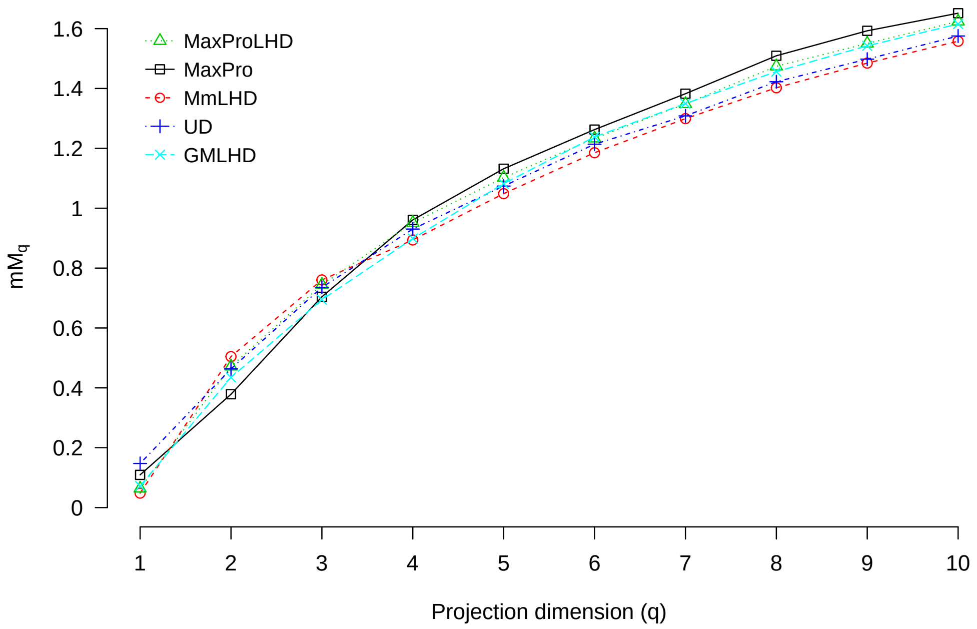

Figure 14: Case 2: mM_q (smaller-the-better)

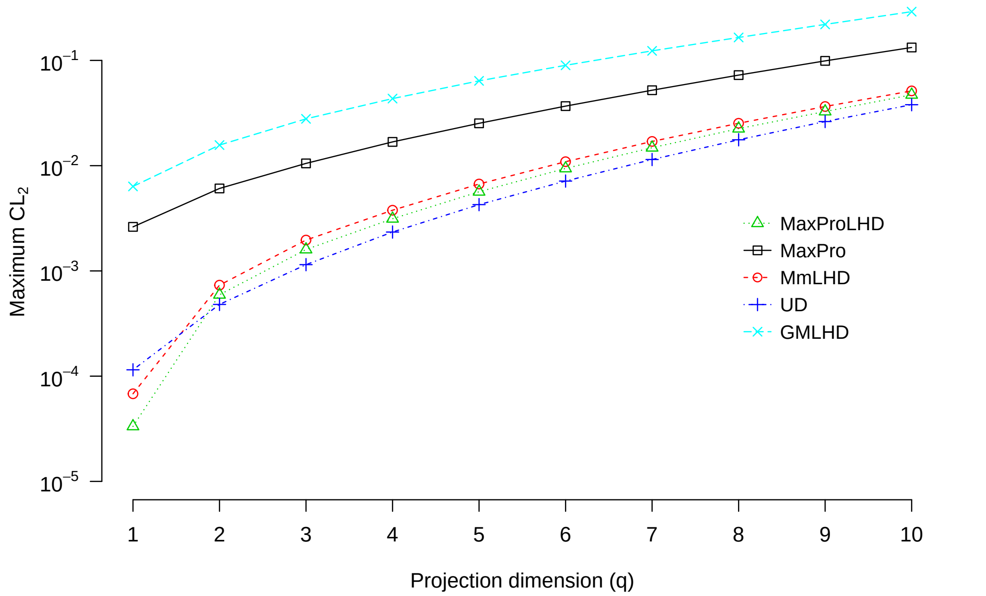

Figure 15: Case 2: Maximum of Centered-L_2-Discrepancies (smaller-the-better)

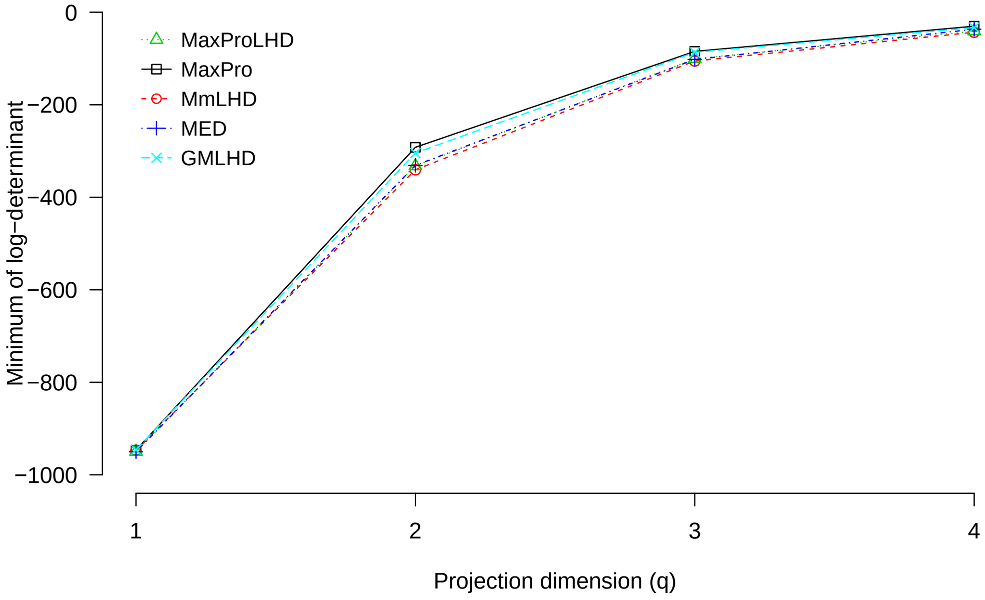

Figure 16: Case 2: Minimum of log-determinant for projection dimensions 1 to 4 (larger-the-better)

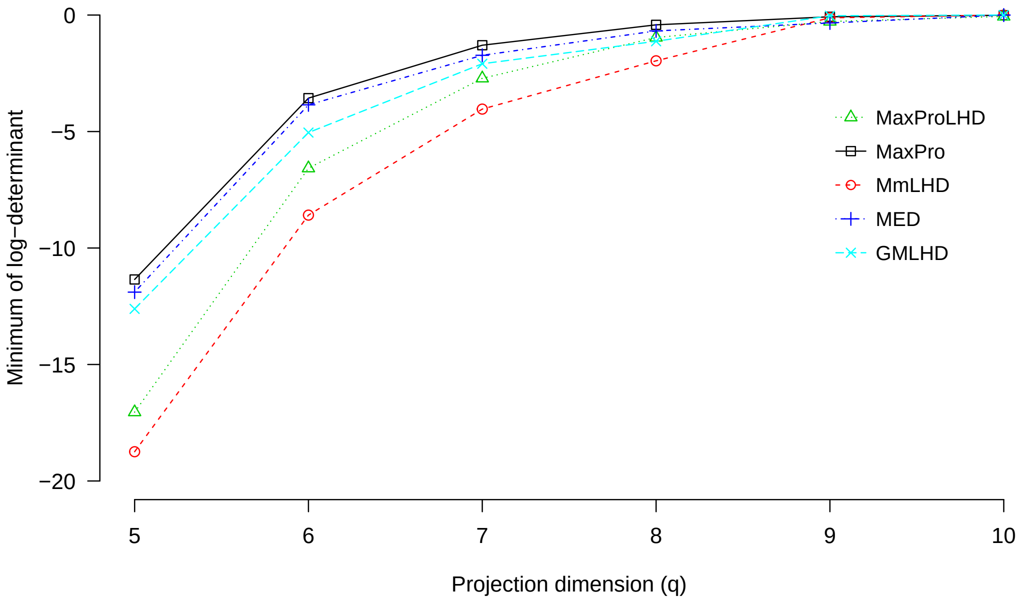

Figure 17: Case 2: Minimum of log-determinant for projection dimensions 5 to 10 (larger-the-better)

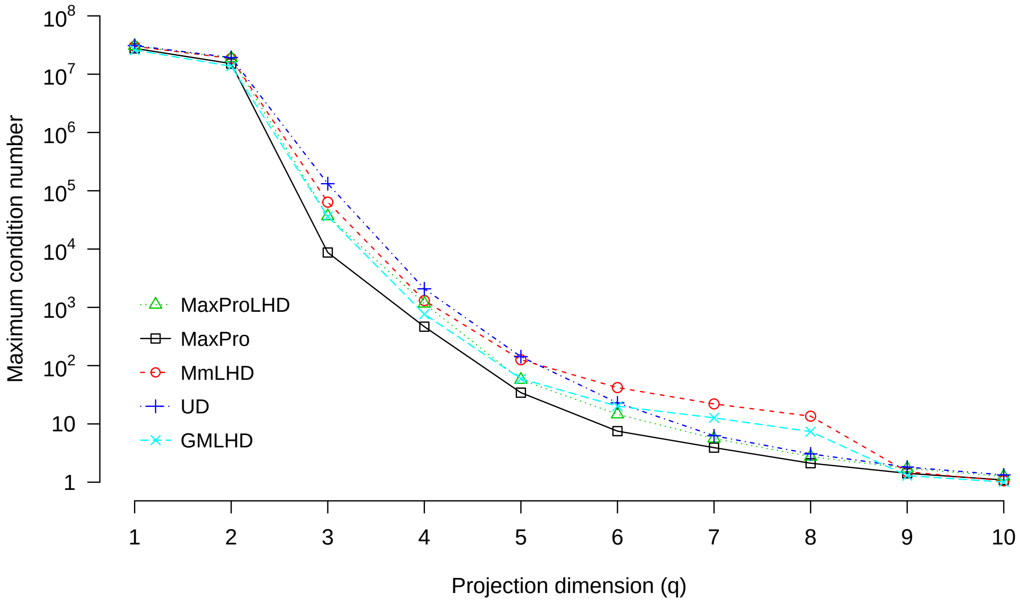

Figure 18: Case 2: Maximum of condition numbers of $\boldsymbol{R}(\boldsymbol{\alpha})$ (smaller-the-better)

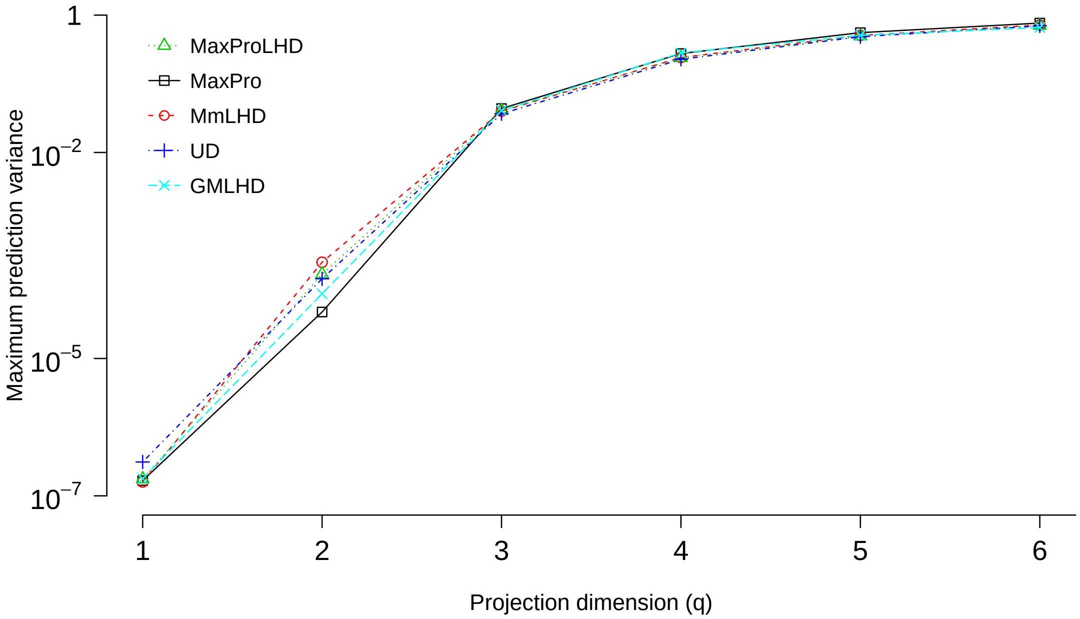

Figure 19: Case 2: Maximum of prediction variance of ordinary kriging (smaller-the-better)

1.7.3 Numerical results for case 3: p=10 and n=15p=150

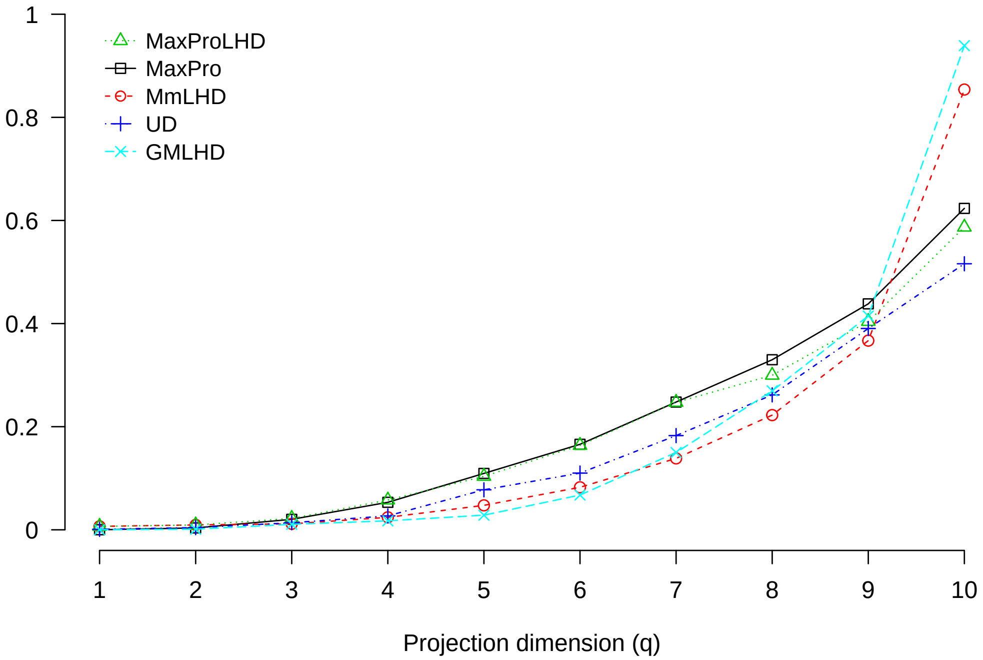

Figure 20: Case 3: Minimum of minimum distances (larger-the-better)

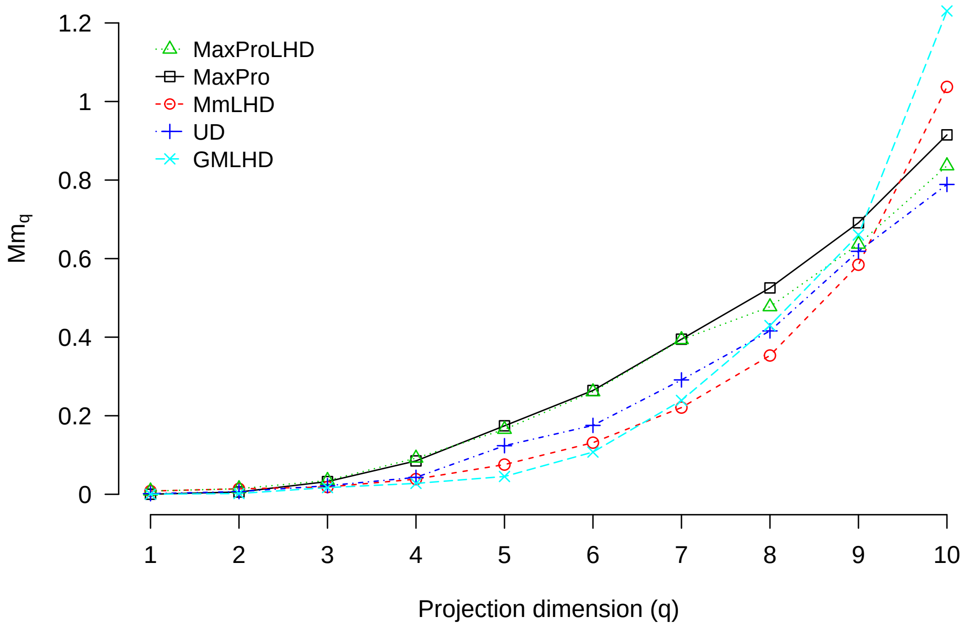

Figure 21: Case 3: Mm_q (larger-the-better)

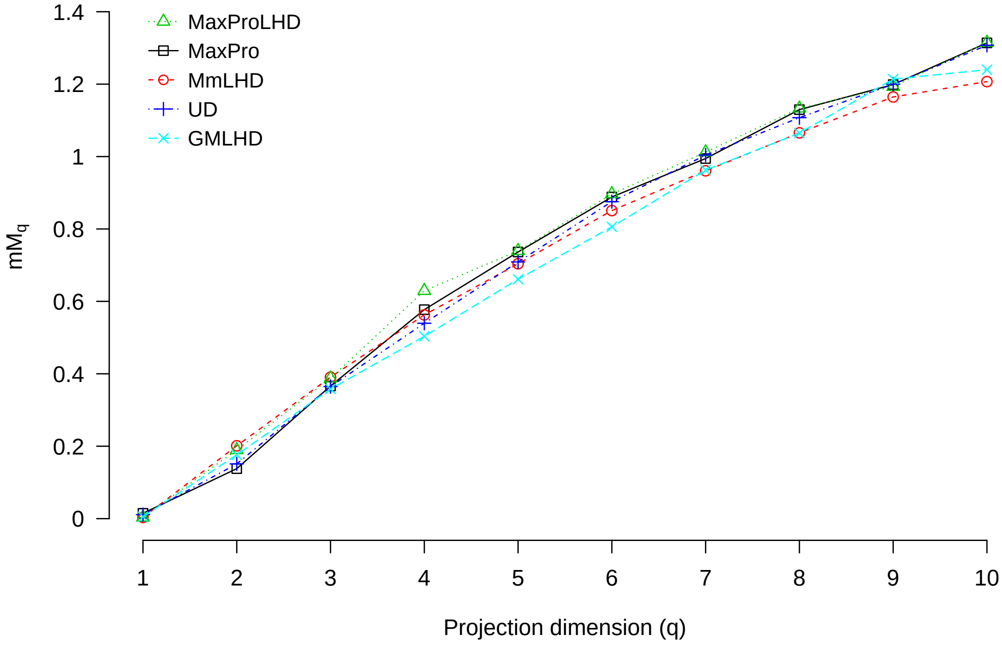

Figure 22: Case 3: Maximum of maximum distances (smaller-the-better)

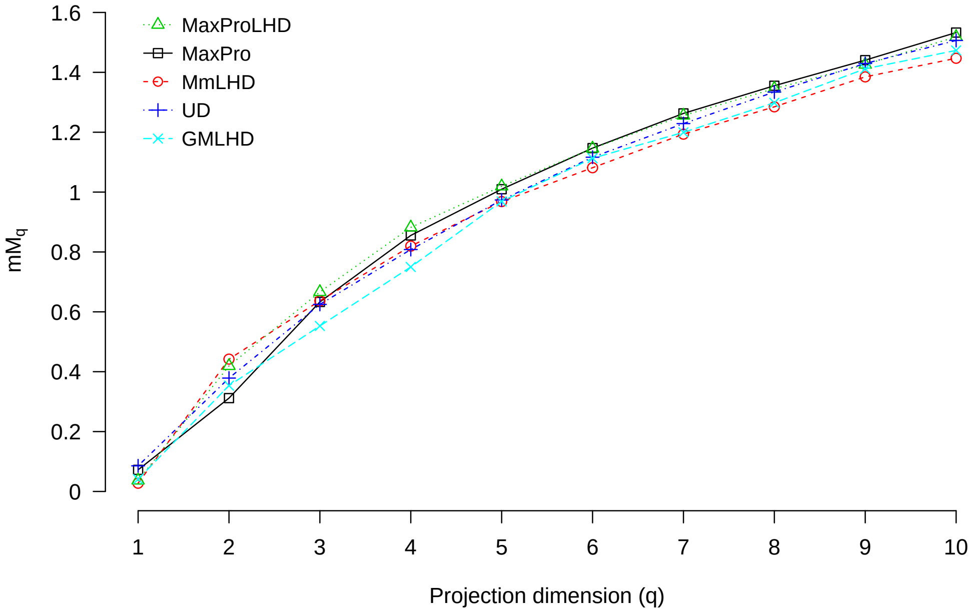

Figure 23: Case 3: mM_q (smaller-the-better)

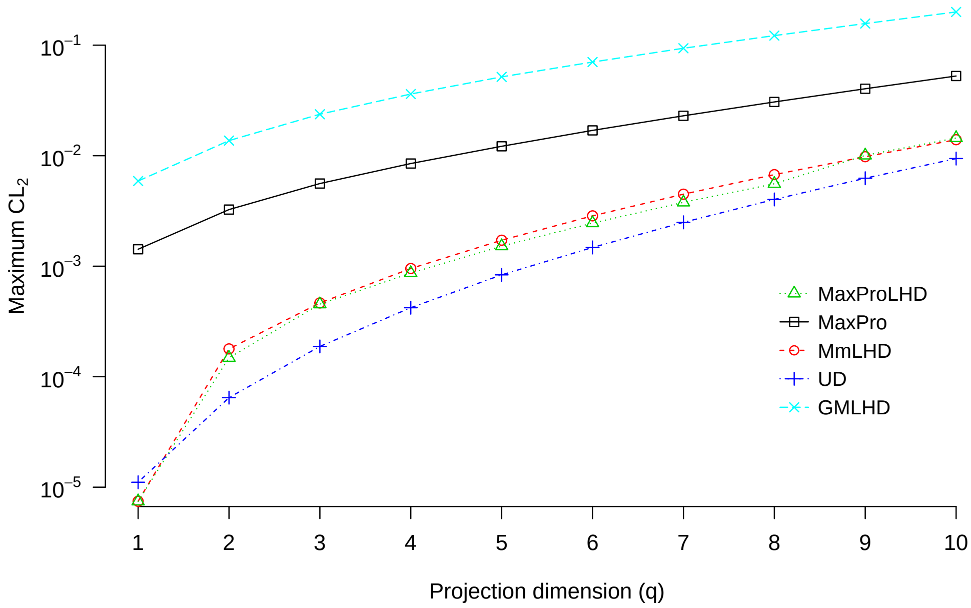

Figure 24: Case 3: Maximum of Centered-L_2-Discrepancies (smaller-the-better)

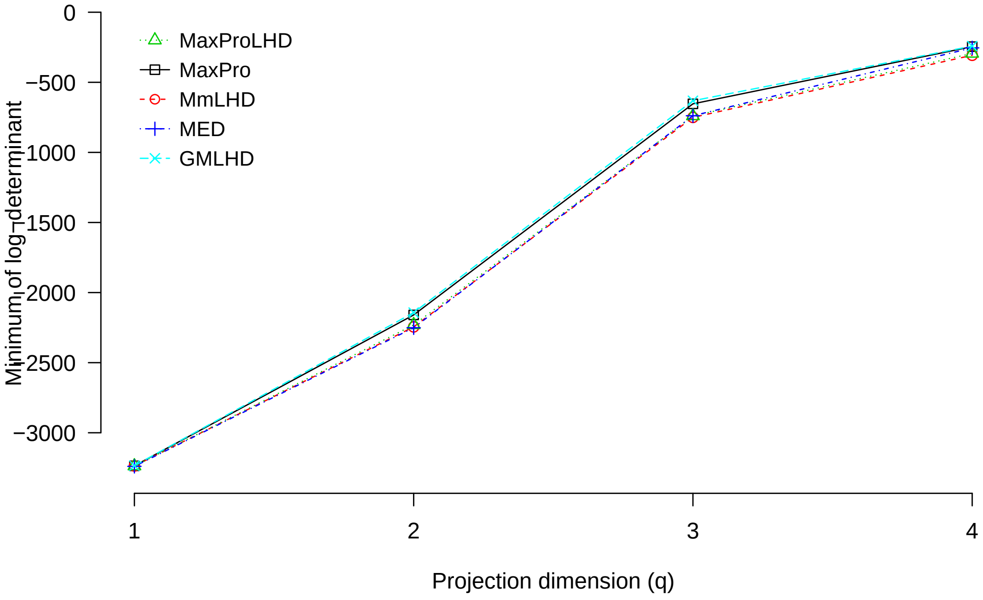

Figure 25: Case 3: Minimum of log-determinant for projection dimensions 1 to 4 (larger-the-better)

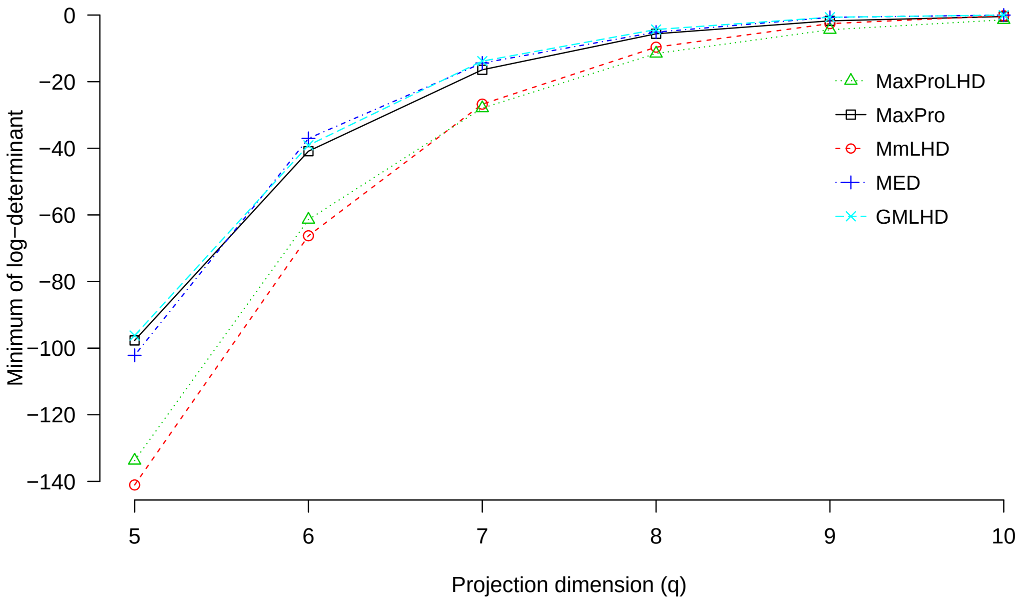

Figure 26: Case 3: Minimum of log-determinant for projection dimensions 5 to 10 (larger-the-better)

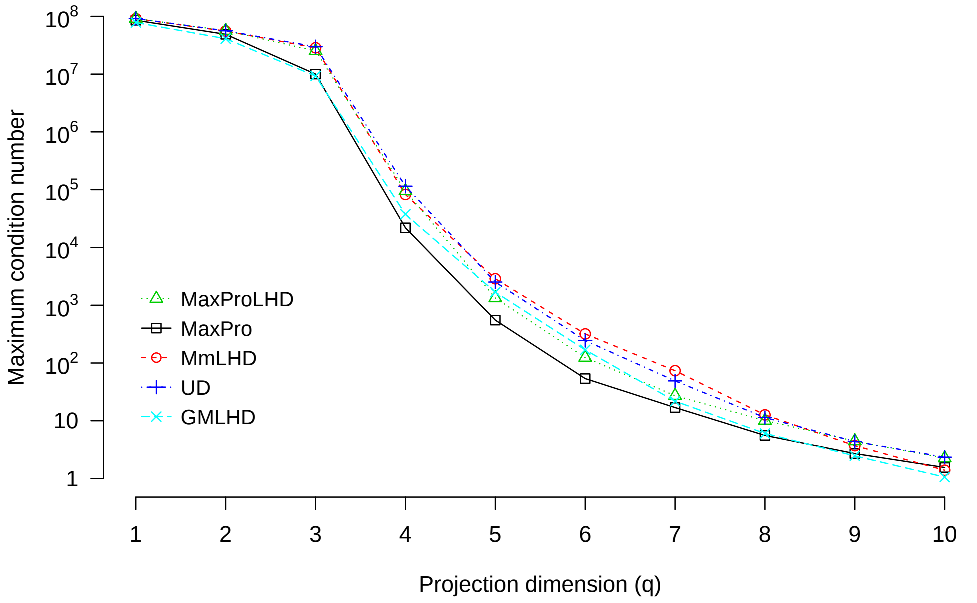

Figure 27: Case 3: Maximum of condition numbers of $\boldsymbol{R}(\boldsymbol{\alpha})$ (smaller-the-better)

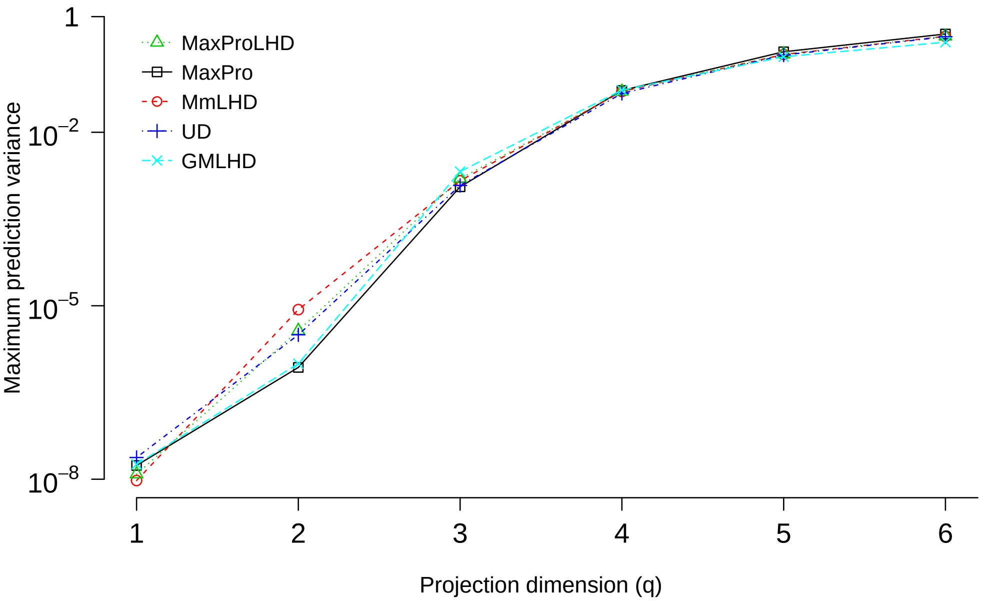

Figure 28: Case 3: Maximum of prediction variance of ordinary kriging (smaller-the-better)

CHAPTER II

UNCERTAINTY QUANTIFICATION IN MACHINING SIMULATIONS USING IN SITU EMULATOR

2.1 Introduction

Uncertainty is all but certain in machining processes. A machining process is subject to uncertainty from various sources such as the inherent random variation in the material and tool geometry. In general, uncertainties in machining simulation fall into two main categories: aleatory and epistemic. Aleatory uncertainty which is irreducible arises naturally from inherent variation and randomness in the system and its environment. On the other hand, epistemic uncertainty occurs due to lack of knowledge about the system and is reducible (Parry 1996). Understanding and identifying the uncertainties in a system is an important step before starting further analysis. In this work, our focus is on the aleatory uncertainty in machining processes.

This work is motivated by the uncertainty quantification problem in a solid end milling simulation which is performed using a deterministic computer model of the Production Module software of Third Wave Systems. This software takes as input the material properties, tool geometry, cutting conditions, and a tool path, and simulates the cutting force components as output. Process is intrinsically characterized with user specified material and tool geometry parameters for a given tool path and workpiece material. These process-specific parameter values are called nominal values. First, process is simulated at the nominal values of the input parameters to obtain the nominal output. However, in reality, output of the process could deviate from the nominal output due to the uncertainties in some of the input parameters, such as hardness of the material, rake angle, helix angle, relief angle, and corner radius. Uncertainties in these parameters are characterized with probability distributions. In this problem, the objective is to quantify the uncertainty in peak tangential force due to the uncertainty in the input parameters. Figure 29 shows the nominal output and four realizations of the peak tangential force. As seen in Figure 29 peak tangential forces vary around the nominal output. Basically, uncertainties in inputs propagate to the output. In a solid end milling process, uncertainty quantification is critical for decision making problems such as tool path optimization, identification of potential areas for improvement and determination of power requirements which are also valid for other machining processes. Therefore, for better informed decision making, reliable and efficient uncertainty quantification is of great importance.

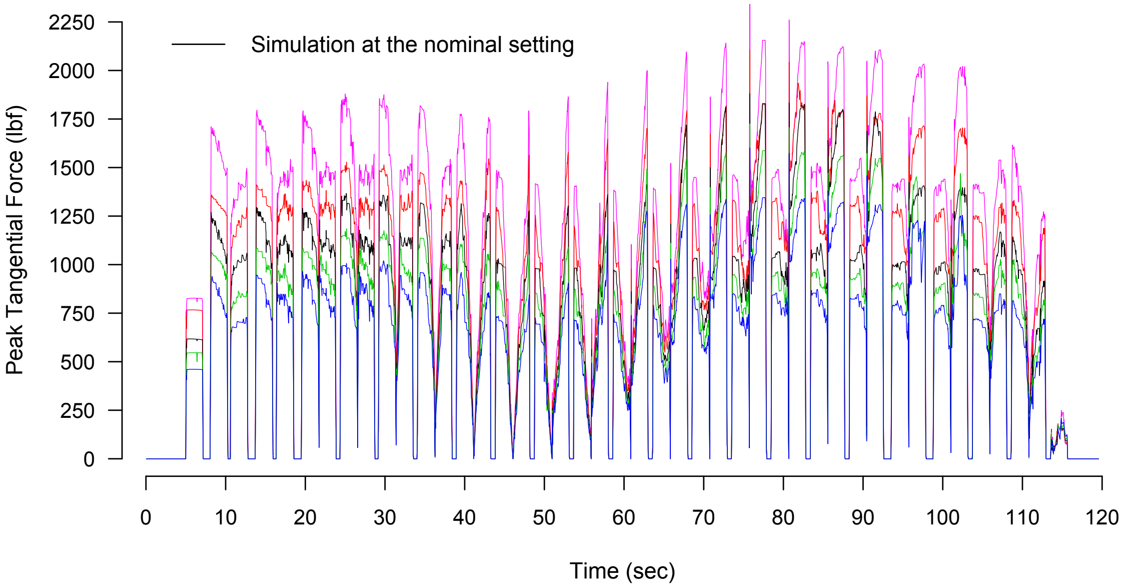

Figure 29: Four realizations of the peak tangential force and the peak tangential force at the nominal setting

Uncertainty quantification methods can be classified into two categories: intrusive and non-intrusive, according to their fundamental approach. Intrusive methods formulate and solve the stochastic version of the computer model to account for uncertainties. Some intrusive methods include perturbation method (Tang and Pinder 1977; Dettinger and Wilson 1981), Neumann series expansion (Zeitoun and Braester 1991) and stochastic Galerkin method (Ghanem and Spanos 1991). Intrusive methods can provide physical interpretation and customization since they are directly applied on the original mathematical model underlying the computer model. However, in real life, implementation of intrusive methods is a challenging and computationally intensive task because of the requirements for extensive modifications and verification of the codes.

On the other hand, non-intrusive methods consider the computer model as a black box and only require multiple solutions of the computer model. An intuitive and simple non-intrusive method for uncertainty quantification is the well known Monte Carlo (MC) method. MC method collects a sample of outputs by running random realizations of the inputs on the computer model and further, extracts the relevant statistics from the sample of outputs. A major drawback of MC is its slow convergence rate. In other words, especially in high dimensional cases, a large number of runs (run size > 1000) should be evaluated to obtain meaningful statistical measures of the output. Considering the fact that a machining simulation takes seconds to hours to even days, the MC method can take several hours or days for even a simple machining process simulation. More intelligent sampling techniques such as Quasi Monte Carlo (QMC) (Fang, Li and Sudjianto 2006) can decrease the computational burden. Using low discrepancy sequences with better uniformity, QMC requires much less simulations for convergence. Still, for a computationally expensive computer model, QMC may become a computational bottleneck. For a comprehensive understanding of sampling-based methods for uncertainty propagation see Helton and Davis (2003) and Helton, Davis and Sallabery (2006).

Despite their simplicity, sampling-based methods end in the grim reality of evaluating the computer model for a prohibitively large number of times. One way to alleviate this problem is to replace the computer model with an easy-to-evaluate metamodel or emulator. Differential analysis (Tomovic and Vukobratovic 1978; Lewins and Becker 1982; Rabitz, Kramer and Dacol 1983; Ronen 1988; Turanyi 1990) and response surface methodology (Kim, Lee, Kim and Chang 1986; Lee, Kim and Chang 1987; Aceil and Edwards 1991) are two of the conventional metamodeling approaches. However, these approaches may not work well if the system is higly nonlinear and the input uncertainty is large. Kriging (Sacks, Welch, Mitchell and Wynn 1989), on the other hand, is a modern tool for metamodeling; it utilizes Gaussian processes to capture nonlinearity in the systems. Kriging emulators have been very popular and used in different application areas for uncertainty quantification (Dwight and Han 2009; DiazDelaO, Adhikari, Flores and Friswell 2013; Xia and Tang 2013; Wana, Mao, Todd and Ren 2014).

Emulators are only an approximation of the computer model. Therefore, for reliable uncertainty quantification, accurate emulators have to be built. Classical approach is to build an emulator for the whole computer model. However, due to the broad region defined by the large number of variables, building generic emulators for machining simulations is a difficult task. Consider the task of building an emulator for the Production Module as an example. Hundreds of material types and tens of tool types with inherent tool geometry parameters define a very large design region. Building an emulator for this region requires extensive number of simulations and computational effort. Moreover, fitting an emulator for the whole region can be inefficient because in a machining process, uncertainty quantification is only needed for the local region defined by the uncertain parameters around the instance which characterizes the process. Thus, a good alternative is to fit an emulator for the local region defined by the given process for which uncertainty quantification is requested. We call it "in situ emulator". It is constructed locally around user-specified nominal and uncertainty values.

2.2 In Situ Emulator Methodology for Uncertainty Quantification

For a given tool path and input parameters with no uncertainties, let

$$y = f(\boldsymbol{u}, t),$$

where y is the output of the computer model, $\boldsymbol{u} = (u_1, u_2 \ldots, u_p)'$ is a vector of p parameters with uncertainties, and $t \in \{t_1, t_2, \ldots, t_M\}$ denote the time points. The time points with either axial depth of cut (ADoC) or radial depth of cut (RDoC) equal to 0 can be eliminated because there is no cutting at those time points. Without loss of generality, let $t = 1, 2, \ldots, m$, where m is the number of time points left after the elimination. It is important to note that in the proposed in situ emulator methodology, we do not consider modeling the output as a function of input parameters with no uncertainties. Since the emulator is built for a user specified process, the input parameters stay fixed at their specified values. Therefore, we only need to model the output with respect to the input parameters with uncertainty. This significantly simplifies the modeling task.

Let the uncertainties in $\boldsymbol{u}$ be characterized with known probability distribution $p(\boldsymbol{u})$ and let $\boldsymbol{u}_0$ be the nominal values of these parameters. Obviously, $\boldsymbol{u}_0$ represents a center value of the probability distribution $p(\boldsymbol{u})$ such as the mean, median, or mode. The objective of this study is to quantify the uncertainties in y caused by the uncertainties in $\boldsymbol{u}$.

Ordinary kriging is a widely used emulator, which is given by

$$y = \mu + Z_t(\boldsymbol{u}), \tag{17}$$

where μ is the overall mean and $Z_t(\boldsymbol{u})$ is a Gaussian process (GP) with mean zero and stationary covariance function $\sigma_t^2 R_t(\cdot)$ (Santner, Williams and Notz 2003). We assume $Z_t(\boldsymbol{u})$ to be independent over time. A more common approach in the literature to model functional responses is to assume a separable covariance function for $\boldsymbol{u}$ and t but with a constant variance σ^2 (Hung, Joseph and Melkote 2015). Our initial attempt to model this process followed this standard approach, but we found that it requires higher computational cost and leads to a predictor with less accuracy. The only problem of ignoring the correlations over time is that we will not be able to predict the output at time points other than the observed time points, but this is not a disadvantage in our case because the output is observed at a very large number of time points and we only need to predict the outputs at those time points.

In metamodeling of machining simulations, instead of using a constant mean, we can develop an emulator which incorporates the output of the simulation at $\boldsymbol{u}_0$. In practice, one first obtains the nominal output by simulating the process at the nominal setting $\boldsymbol{u}_0$, and then requests uncertainty analysis if needed. Thus, the output of the simulation at $\boldsymbol{u}_0$ is always available. In general, especially when the uncertainty is not large, $\boldsymbol{u}$ takes a value around its nominal value with a high probability. Therefore, we can expect output values to have similar trend with the ones obtained from the nominal setting, $\boldsymbol{u}_0$. For example, in Figure 29, trends of the realizations and the nominal output match well for the solid end milling process. Thus, we propose

$$\psi(f(\boldsymbol{u}, t)) = \psi(f(\boldsymbol{u}_0, t)) + \delta_t(\boldsymbol{u}), \tag{18}$$

where $\psi(\cdot)$ is an appropriate transformation on the output, $f(\boldsymbol{u}_0, t)$ is the evaluation of the computer model at the nominal setting $\boldsymbol{u}_0$ and $\delta_t(\boldsymbol{u})$ is a function which captures the discrepancy between $\psi(f(\boldsymbol{u}, t))$ and $\psi(f(\boldsymbol{u}_0, t))$. Since there is no discrepancy at the nominal setting $\boldsymbol{u}_0$,

$$\delta_t(\boldsymbol{u}_0) = 0 \text{ for all } t \in \{1, 2, \ldots, m\}. \tag{19}$$

In formulation (18), a transformation is considered because we expect a suitable transformation on the response would scale down the discrepancy term substantially for simpler modeling. Models for machining processes are usually sophisticated, therefore, discrepancy term must be properly specified. To satisfy (19), we let

$$\delta_t(\boldsymbol{u}) = Z_t(\boldsymbol{u}) - Z_t(\boldsymbol{u}_0), \tag{20}$$

where $Z_t(\boldsymbol{u})$ and $Z_t(\boldsymbol{u}_0)$ are GPs with correlation function $R(\cdot)$. Thus, $\delta_t(\boldsymbol{u})$ is also a GP with mean 0 and covariance function $\sigma_t^2 R(\cdot)$ where correlation function is given by

$$\begin{aligned} R_t(\boldsymbol{u}_i - \boldsymbol{u}_j) &= corr\left(\delta_t(\boldsymbol{u}_i), \delta_t(\boldsymbol{u}_j)\right) \\ &= corr\left(Z_t(\boldsymbol{u}_i) - Z_t(\boldsymbol{u}_0), Z_t(\boldsymbol{u}_j) - Z_t(\boldsymbol{u}_0)\right), \\ &= R_{t,\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_j) - R_{t,\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_0) - R_{t,\boldsymbol{u}}(\boldsymbol{u}_j - \boldsymbol{u}_0) + 1. \end{aligned} \tag{21}$$

Commonly used Gaussian correlation function is chosen for $R_{t,\boldsymbol{u}}(\cdot)$,

$$R_{t,\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_j) = \exp\{-\sum_{k=1}^{p} \theta_{t,k}(u_{ik} - u_{jk})^2\}.$$

The use of a nominal function in (18) and the nonstationary correlation function in (21) are two novel parts of our proposed in situ emulator methodology.

Suppose we have an n-run p-factor experimental design $\boldsymbol{D}_{\boldsymbol{u}} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$. For the emulator in (18), we define the output as a vector $\boldsymbol{y}_t = (y_{1,t}, \ldots, y_{n,t})$ with j^{th} element $y_{jt} = f(\boldsymbol{u}_j, t)$. In addition, for the nominal setting, $\boldsymbol{y}_{0,t} = (y_{0,t}, \ldots, y_{0,t})$ by replicating $y_{0,t} = f(\boldsymbol{u}_0, t)$ n times. Lastly, let the transformed outputs be $\boldsymbol{w}_t = \psi(\boldsymbol{y}_t)$ and $\boldsymbol{w}_{0,t} = \psi(\boldsymbol{y}_{0,t})$. Then, the correlation parameters $\boldsymbol{\theta}_t$, $t = 1, \ldots, m$, are found by minimizing the negative log-likelihood function [?]

$$\sum_{t=1}^{m} \{n \log \hat{\sigma}_t^2 + \log |\boldsymbol{R}_t|\}, \tag{22}$$

where $\boldsymbol{R}_t$ is an $n \times n$ matrix with ij^{th} element $R_t(\boldsymbol{u}_i - \boldsymbol{u}_j)$ and $\hat{\sigma}_t^2 = \frac{1}{n}(\boldsymbol{w}_t - \boldsymbol{w}_{0,t})' \boldsymbol{R}_t^{-1} (\boldsymbol{w}_t - \boldsymbol{w}_{0,t})$. In a machining simulation, m can range from thousands to millions. Therefore, optimization problem in (22) can be very time-consuming which makes uncertainty quantification of the machining process inefficient. To simplify the computation, we assume that the effect of a parameter is independent of time. Therefore, we let $\boldsymbol{\theta}_t = \boldsymbol{\theta}$ for all $t \in \{1, 2, \ldots, m\}$. Now, the common set of correlation parameters, $\boldsymbol{\theta}$, can be found by minimizing,

$$n \sum_{t=1}^{m} \log \hat{\sigma}_t^2 + m \log |\boldsymbol{R}|, \tag{23}$$

where $\boldsymbol{R}$ is an $n \times n$ matrix with elements calculated using $\boldsymbol{\theta}$ and $\hat{\sigma}_t^2 = \frac{1}{n}(\boldsymbol{w}_t - \boldsymbol{w}_{0,t})' \boldsymbol{R}^{-1} (\boldsymbol{w}_t - \boldsymbol{w}_{0,t})$.

The discrepancy function is given by

$$\hat{\delta}_t(\boldsymbol{u}) = \boldsymbol{r}(\boldsymbol{u})' \boldsymbol{R}^{-1} (\boldsymbol{w}_t - \boldsymbol{w}_{0,t}), \tag{24}$$

where $\boldsymbol{r}(\boldsymbol{u}) = (R(\boldsymbol{u} - \boldsymbol{u}_1), \cdots, R(\boldsymbol{u} - \boldsymbol{u}_n))'$. Thus, the in situ emulator is obtained by substituting (24) into (18),

$$\hat{f}(\boldsymbol{u}, t) = \psi^{-1}[\psi(f(\boldsymbol{u}_0, t)) + \boldsymbol{r}(\boldsymbol{u})' \boldsymbol{R}^{-1} (\boldsymbol{w}_t - \boldsymbol{w}_{0,t})]. \tag{25}$$

Succinctly, predictions for the whole process can be calculated by

$$\hat{\boldsymbol{f}}(\boldsymbol{u}) = \psi^{-1}[\psi(\boldsymbol{f}(\boldsymbol{u}_0)) + \boldsymbol{r}(\boldsymbol{u})' \boldsymbol{R}^{-1} \boldsymbol{W}], \tag{26}$$

where $\hat{\boldsymbol{f}}(\boldsymbol{u})$ and $\boldsymbol{f}(\boldsymbol{u}_0)$ are vectors of length m with j^{th} elements equal to $\hat{f}(\boldsymbol{u}, t_j)$ and $f(\boldsymbol{u}_0, t_j)$, respectively, and $\boldsymbol{W}$ is an $n \times m$ matrix with j^{th} column equal to $\boldsymbol{w}_j - \boldsymbol{w}_{0,j}$.

Uncertainty quantification of a machining process requires predictions on a large set of simulation settings. Let $\boldsymbol{D}_{uq}$ is an $N \times p$ matrix with each row corresponding to a simulation setting, then, predictions for N settings can be done efficiently by

$$\hat{\boldsymbol{F}}_{uq} = \psi^{-1}[\psi(\boldsymbol{F}_0) + \boldsymbol{R}_{uq}\boldsymbol{R}^{-1}\boldsymbol{W}], \tag{27}$$

where $\boldsymbol{F}_0$ is an $N \times m$ matrix with each row equal to $\boldsymbol{f}(\boldsymbol{u}_0)$, $\boldsymbol{R}_{uq}$ is an $N \times n$ matrix with j^{th} row equal to $\boldsymbol{r}(\boldsymbol{u}_j)$, and $\psi(\cdot)$ is applied on each element of the matrix.

2.3 Uncertainty Quantification in Solid End Milling Processes

End milling is one of the most commonly used cutting processes for precise machining of complicated parts in the aerospace industry. In this section, first, we illustrate the approach and carry out an uncertainty analysis on the 3-axis solid end milling process in Figures 30 and 31 with the proposed methodology, which is based on the widely used Production Module software of Third Wave Systems. Later, our approach is also applied on a complex 5-axis cutting process simulated in Production Module as well. Our objective is to quantify the uncertainty in the output arising from uncertain input parameters. Production Module simulates the solid end milling process with a computer model based on physics-based material models. In the computer model, the user defines the process by setting the several material and tool properties. Some of these properties are hardness of the material, cutter diameter, rake angle, helix angle, relief angle, corner radius, number of flutes, flute length and tool length. In the process, process variables, feed rate, radial depth of cut (RDoC) and axial depth of cut (ADoC) vary as a function of time but they remain the same for any input setting. The computer model is deterministic and it produces a set of outputs for a specified setting and tool path. Peak tangential force is one of the key outputs in the solid end milling process, which is studied in this application. Finally, the material used in the process is AISI-1020 steel.

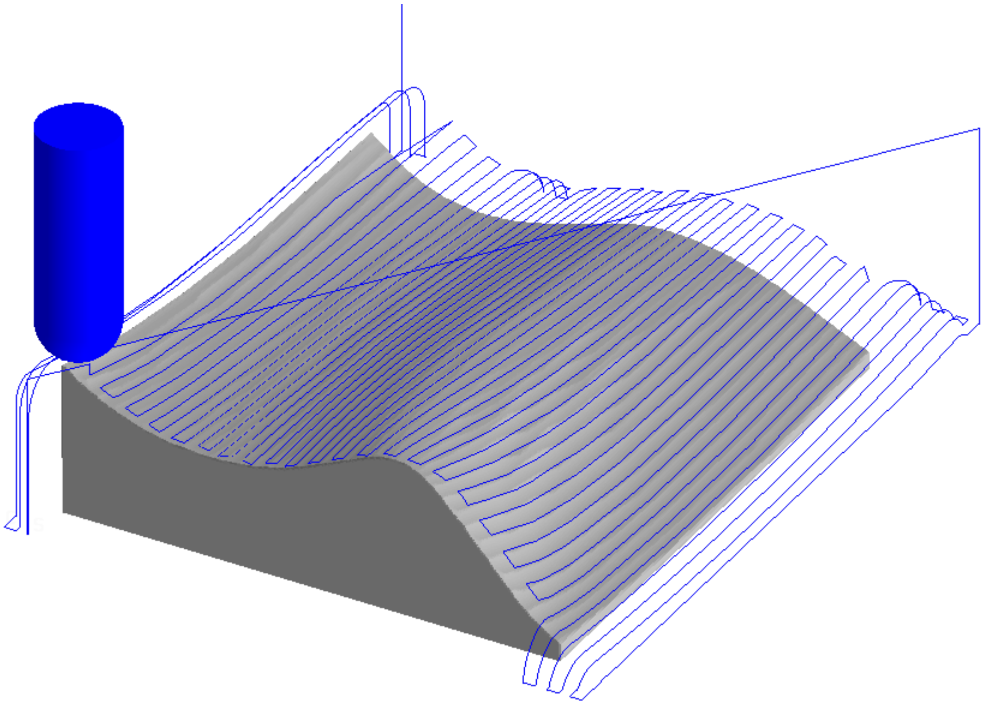

Figure 30: The cutting tool, toolpath and the workpiece

In the process, four input parameters are identified with uncertainty: hardness of the material (u_1), rake angle (u_2), helix angle (u_3), and corner radius (u_4). Table 1 provides nominal values and the probability distributions of the four input parameters. We specify hardness of material and helix angle to follow a normal distribution with means equal to their nominal values and standard deviations equal to 15% of their nominal values. Rake angle is assumed to vary around its nominal value with exponential distribution with the distribution parameter $\lambda = 100$. Last, corner radius comes from a beta distribution, which has the distribution parameters $\alpha = 50$ and $\beta = 1$, scaled by 0.5 to comply with the range of corner radius. These distributions can be changed by the user depending on the particular process and problem under consideration. As seen in Figure 31, the computer simulation spans around two minutes over which 3373 force values are collected.

The in situ emulator is given by

$$\log f(\boldsymbol{u}, t) = \log f(\boldsymbol{u}_0, t) + \delta_t(\boldsymbol{u}), \tag{28}$$

where $\boldsymbol{u}$ can vary around $\boldsymbol{u}_0$ based on its specified probability distribution and all

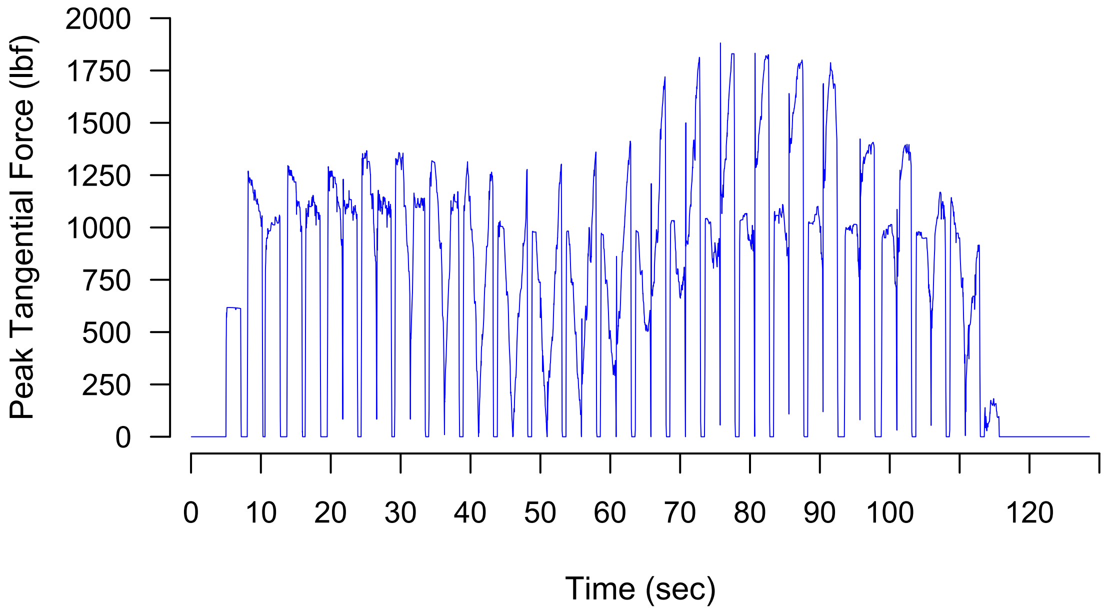

Figure 31: Peak tangential forces over time at the nominal setting

other input parameters are held constant at their specified values. We use log-transformation in (28) because the cutting force can usually be approximated by a multiplicative functional form of input variables which makes log-transformation an appropriate transformation for the output.

Design of experiments is a critical step for obtaining an accurate emulator. We use space-filling designs which are commonly used in deterministic computer experiments. In practice, it is highly likely that only a small set of inputs are significant (Wu and Hamada 2009), therefore, it is preferable to adopt a design which is space filling for all possible input combinations. We employ maximum projection designs (Joseph, Gul and Ba 2015) for the experimentation since they are both space filling in the original design space, spanned by all input parameters and also in the lower dimensional projections, spanned by subsets of input parameters. We generate 10 experimental settings which are evenly spread in the full design space and around the nominal

Table 1: Nominal values and the probability distributions of the input parameters with uncertainty

Parameter	Nom. Value	Probability Dist.
u_1:Hardness	111 Bhn	$N(111, (0.15 \times 111)^2)$
u_2:Rake Angle	0 deg	$Exp(\lambda = 100)$
u_3:Helix Angle	20 deg	$N(20, (0.15 \times 20)^2)$
u_4:Corner Radius	0.5 in	$Beta(\alpha = 50, \beta = 1)/2$

Table 2: 11-run 4-factor experimental design

Hardness	Rake Ang.	Helix Ang.	Corner Rad.
u_1	u_2	u_3	u_4
111	0	20	0.5
82.85	0.001	21.42	0.492
92.74	0.005	16.71	0.497
98.55	0.014	23.29	0.499
103.13	0.006	20.70	0.470
107.17	0.020	18.58	0.494
114.83	0.009	14.93	0.489
118.87	0.003	25.07	0.495
123.45	0.031	22.24	0.486
129.26	0.002	17.76	0.481
139.15	0.011	19.31	0.498

setting to maximize the information collected about the process. Accordingly, a 11-run 4-factor maximum projection Latin hypercube design is generated in $[0, 1]^4$ with the first run fixed at the nominal setting. Since maximum projection Latin hypercube design is approximately uniform (Joseph, Gul and Ba 2015), the design in the original input space is obtained with inverse probability transformation and it is given in Table 2.

After the experimentation, the time points where there are no cuttings ($RDoC = ADoC = 0$) are eliminated and we are finally left with $m = 2283$ time points. The correlation parameters are found as $\hat{\boldsymbol{\theta}} = (0.416, 0.019, 4.569, 0.031)'$ by minimizing (23). Now, the peak tangential forces can be predicted using

$$\hat{\boldsymbol{f}}(\boldsymbol{u}) = \exp[\log(\boldsymbol{f}(\boldsymbol{u}_0)) + \boldsymbol{r}(\boldsymbol{u})'\boldsymbol{R}^{-1}\boldsymbol{W}]. \tag{29}$$

Prediction performance of the in situ emulator is tested on 40 simulations. Simulation settings are constructed with a 40-run 4-factor uniform design generated using the

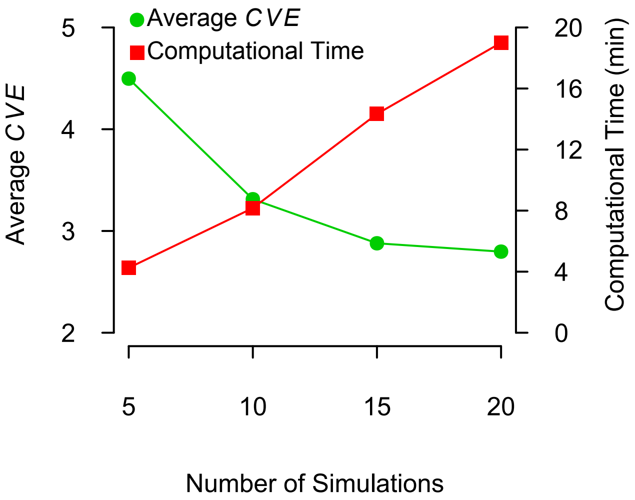

Figure 32: Average CVE and computational time with respect to number of simulations

statistical software JMP 11. The coefficient of variation of the root mean square error (CVE) is used as a performance measure:

$$CVE_j = 100 \times \frac{\sqrt{\sum_{t=1}^{m} (\hat{f}(\boldsymbol{u}_j, t) - f(\boldsymbol{u}_j, t))^2 / m}}{\bar{f}(\boldsymbol{u}_j)} \tag{30}$$

where $\bar{f}(\boldsymbol{u}_j) = \sum_{t=1}^{m} f(\boldsymbol{u}_j, t)/m$ for $j = 1, \ldots, 40$. We found that CVE values have a mean of 3.30. In other words, on average, the force predictions for a setting will have a root mean square error 3.30% of the mean of the forces obtained from the computer model, which is very small. Therefore, the in situ emulator accurately predicts the peak tangential forces.

Modeling with larger designs increases the prediction capability of in situ emulator, however, it also increases the computational time. Figure 32 shows the average CVE values and computational times for different design sizes on a 2.9 GHz computer. As seen in Figure 32, if we double the number of simulations in the current design, average CVE improves 15%, on the other hand, computational time increases more than two times. Thus, in this example, current design with 10 simulations is a good compromise between prediction accuracy and computational efficiency.

We quantify the uncertainty by constructing the confidence interval on the peak tangential force. Confidence interval for the in situ emulator is constructed from the predictions of 40 input settings. We also compute the confidence interval using the QMC method from 40 simulations. Figure 33 shows the 95% confidence intervals on the peak tangential force by both methods. Results show that the confidence interval constructed by the in situ emulator reasonably agrees with that of the QMC method.

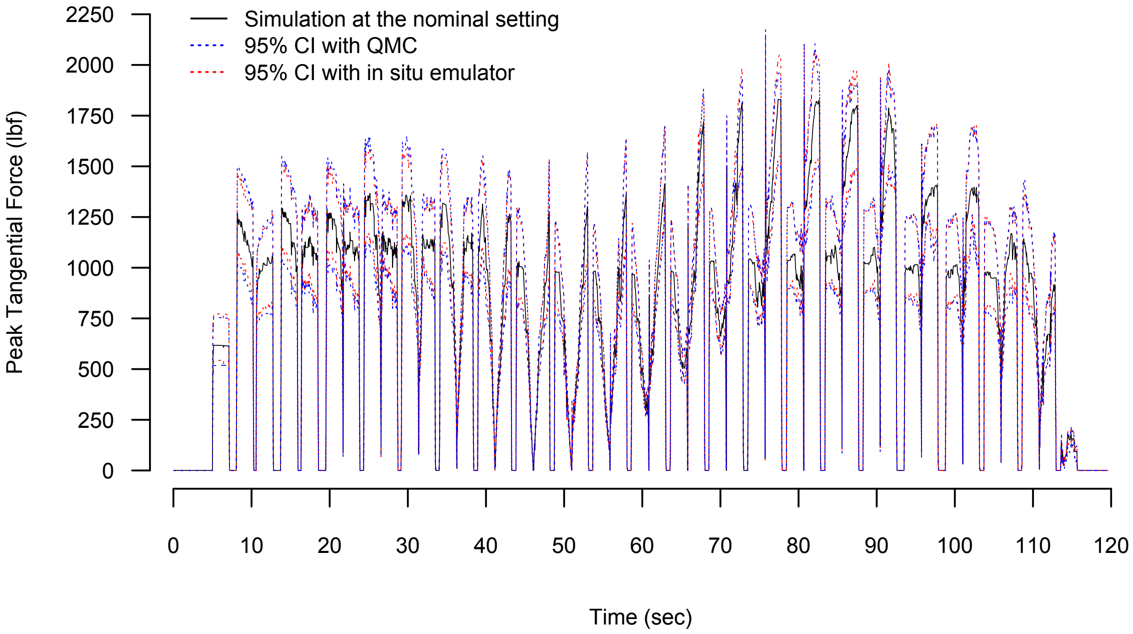

Figure 33: 95% Confidence intervals constructed by QMC and in situ emulator for the solid end milling process

For validation, we perform uncertainty quantification for a more complex solid end milling process. It is a 5-axis CNC cutting process for which cutting tool, toolpath, workpiece, and a portion of the peak radial forces at the nominal setting are shown in Figures 34 and 35. In this process, uncertainty in peak radial force is to be quantified. Hardness of the material (u_1), rake angle (u_2), helix angle (u_3), and corner radius (u_4) are the uncertainty sources for this process. Table 3 provides nominal values

Table 3: Nominal values and the probability distributions of the input parameters with uncertainty for 5-axis CNC cutting

Parameter	Nom. Value	Probability Dist.
u_1:Hardness	111 Bhn	$N(111, (0.15 \times 111)^2)$
u_2:Rake Angle	5 deg	$N(5, (0.15 \times 5)^2)$
u_3:Helix Angle	30 deg	$N(30, (0.15 \times 30)^2)$
u_4:Corner Radius	0.09 in	$N(0.09, (0.15 \times 0.09)^2)$

and the probability distributions of these four input parameters, where all follow normal distribution with means equal to their nominal values and standard deviations equal to 15% of their nominal values. Computer simulation predicts the process to take around 10 minutes and collects $155,107$ force values over time. For a better visualization, Figure 35 shows the first 80 seconds of the computer simulation where $20,000$ force values are plotted. The peak radial force of the whole 5-axis CNC cutting process is made up of cycles as seen in Figure 35.

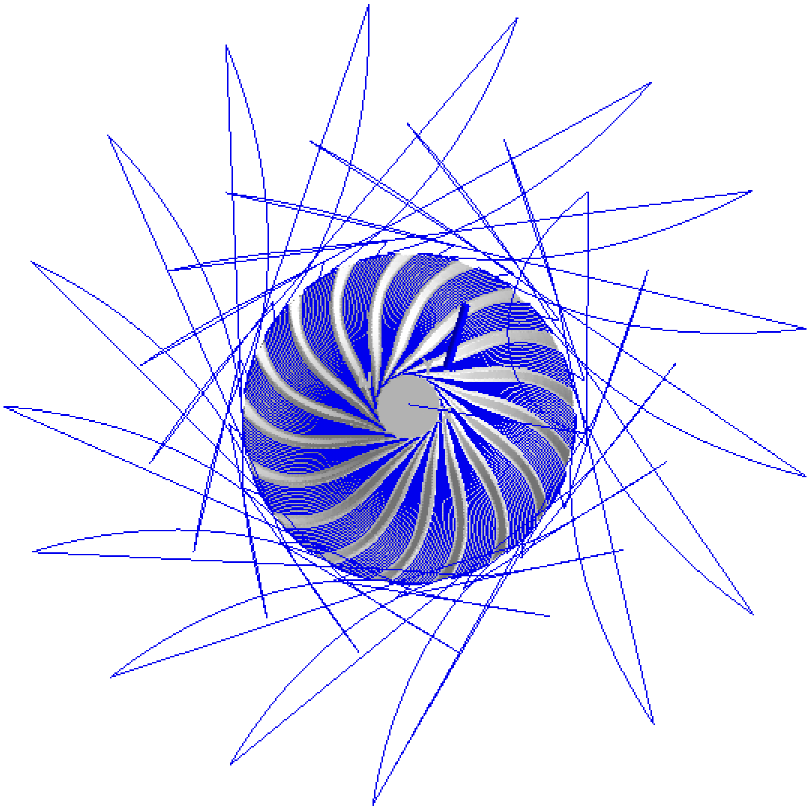

Figure 34: The cutting tool, the toolpath, and the workpiece for 5-axis CNC cutting

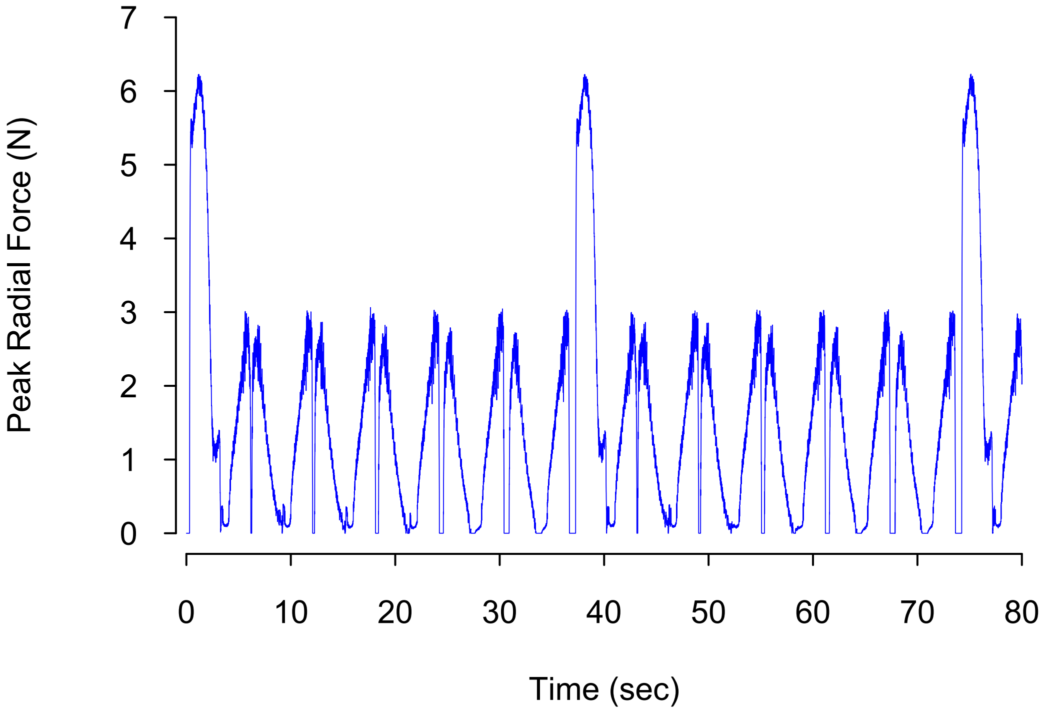

Figure 35: Peak radial force over time for 5-axis CNC cutting

We proceed as before and obtained the in situ emulator for the 5-axis CNC cutting using an 11-run 4-factor maximum projection Latin hypercube design. The mean CVE is found to be 4.10, which shows that the in situ emulator has good prediction accuracy for the 5-axis CNC cutting as well. Figure 36 has the 95% confidence intervals on the peak radial force obtained by the in situ emulator and QMC method using 40 simulations. For the 5-axis CNC cutting as well, we observe a satisfactory match between the confidence intervals by both methods.

In order to obtain reliable results, uncertainty quantification should be done with larger sample of input settings. In practice, uncertainty in output can be quantified with say, 2000 evaluations. In that case, for the 3-axis solid end milling process, uncertainty quantification takes only eight minutes for the in situ emulator on a 2.9 GHz computer. Estimation of the correlation parameters and evaluation of the 2000 settings constitute 45 and 3 seconds of the total eight minutes, respectively. The remaining computational time is the experimentation on the 10 simulation settings.

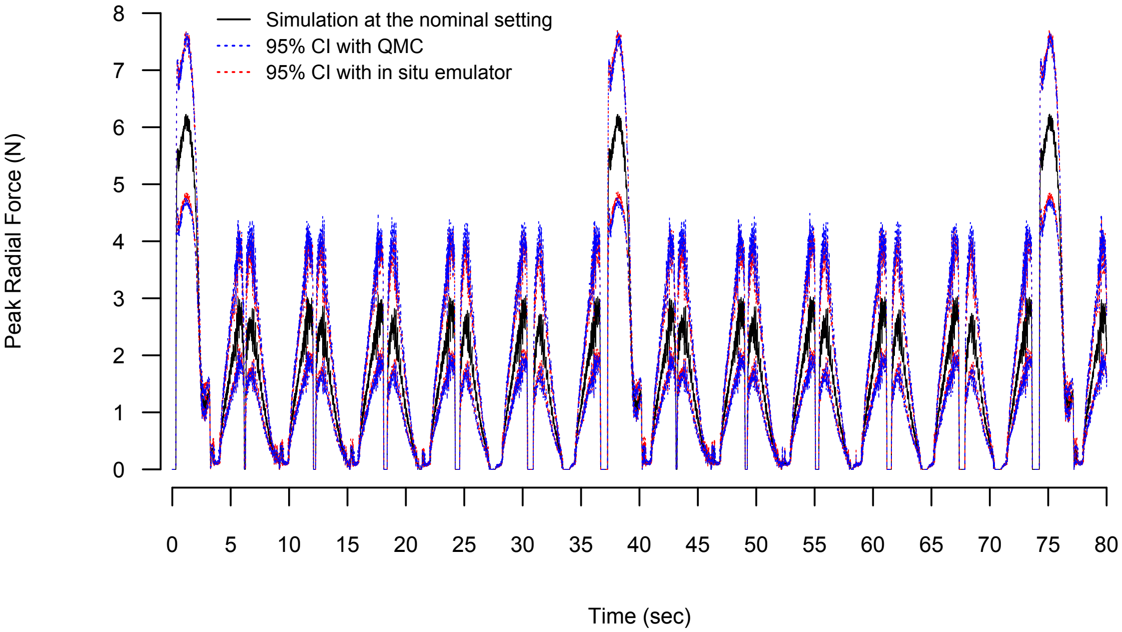

Figure 36: 95% Confidence intervals constructed by QMC and in situ emulator for the 5-axis CNC cutting

On the other hand, uncertainty quantification with 2000 evaluations will take more than a day for the QMC method. For 5-axis CNC cutting, the in situ emulator takes only one hour (53.5 minutes for 10 simulations, 5 minutes is for the estimation of the correlation parameters, and 1.5 minutes for the evaluation of the 2000 settings), whereas the QMC method will require nine days for 2000 simulations. Therefore, the in situ emulator provides huge computational savings for these two applications.

2.4 Conclusions

In this chapter, we present an efficient uncertainty quantification methodology for machining simulations. In real life applications, computer models tend to be complex and high dimensional. Therefore, brute force implementation of sampling-based methods are not affordable. Besides, fitting a comprehensive emulator for machining processes is also not practically feasible. On the other hand, in the proposed approach, we build a local in situ emulator for a given process when an uncertainty quantification is demanded. Because of the local nature, the response surface can be approximated using a few simulations, which makes our approach very fast. On the down side, a new set of simulations are needed for each new process or change in the uncertainty variables and their probability distributions. One approach to overcome this deficiency is to augment the previous set of simulations with a few new simulations, but a complete development of this idea needs further research.

The proposed methodology is applied to two solid end milling processes for the quantification of uncertainty in the force outputs. In the application, the in situ emulators are estimated with only ten more machining simulations over the nominal simulation. Accuracy of the in situ emulators are tested on a set of simulations. According to the results, the in situ emulator is an accurate approach with significant computational savings. Consequently, the in situ emulator is an attractive alternative to sampling-based approaches for the uncertainty quantification of complex machining process simulations.

CHAPTER III

MAXIMUM PROJECTION DESIGNS WITH QUANTITATIVE AND QUALITATIVE FACTORS

3.1 Introduction

Real-word phenomena have been increasingly studied virtually due to the advances in high performance computation and impracticable and infeasible physical experimentation. This *virtual* experiments is called computer experiments. In deterministic computer experiments, space-filling designs are widely used and one of the intuitive and popular approach for achieving space-fillingness is to maximize the minimum pairwise distance in a design (Johnson, Moore and Ylvisaker 1990). Let $\mathcal{X} = [0,1]^p$ be the experimental region defined by p factors. Suppose we want to construct an n-run design $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ where each point $\boldsymbol{x}_i \in \mathcal{X}$. The maximin distance design can be obtained as,

$$\max_{\boldsymbol{D}} \min_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{D}} d(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{31}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between the points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Another approach is to minimize the maximum distance between any point $\boldsymbol{x} \in \mathcal{X}$ and the design. The minimax distance design is defined as,

$$\min_{\boldsymbol{D}} \max_{\boldsymbol{x} \in \mathcal{X}} d(\boldsymbol{x}, \boldsymbol{D}), \tag{32}$$

where $d(\boldsymbol{x}, \boldsymbol{D}) = \min_{\boldsymbol{x}_i \in \boldsymbol{D}} d(\boldsymbol{x}, \boldsymbol{x}_i)$.

Computer models are usually deterministic; there is no random error. Therefore, in computer experiments, replications are unnecessary and a good design should not have replications when projected onto any subspace of the input factors. Although maximin and minimax distance designs are space-filling in $\mathcal{X} = [0,1]^p$, they may have replications when projected onto any factor. Morris and Mitchell (1995) proposed to construct maximin distance designs with Latin hypercube structure which ensures n

equally spaced points for each factor. They introduced the maximin Latin hypercube design (MmLHD) which is given by

$$\min_{\boldsymbol{D}} \quad \phi_k(\boldsymbol{D}) = \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d^k(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right\}^{1/k}, \tag{33}$$

where k is large enough to achieve maximin distance. MmLHD criterion ensures space-fillingness in full dimensional space and uniformity in one dimensional space, however, it does not guarantee good projections onto the subspaces with dimensions $2, \ldots, p-1$. Joseph et al. (2015) proposed *maximum projection designs* (MaxPro) which have good projections for all subspaces of the factors. The MaxPro design is obtained with the criterion:

$$\min_{\boldsymbol{D}} \quad \psi(\boldsymbol{D}) = \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\prod_{l=1}^{p} (x_{il} - x_{jl})^2} \right\}^{1/p}. \tag{34}$$

If two points have the same coordinate for a dimension, MaxPro criterion becomes ∞. Therefore, MaxPro criterion automatically possesses Latin hypercube structure. They also proposed maximum projection Latin hypercube design (MaxProLHD) with equally spaced points in each dimension. MaxPro design tends to put more points towards the boundaries which increases prediction capability in Gaussian process modeling (Dette and Peppelyshev 2010). Therefore, MaxPro design is a better choice for prediction purposes, whereas, MaxProLHD with better uniformity is preferable for integration purposes. Several other works have been also done on finding space-filling designs with different optimality criteria. Some of these space-filling designs are uniform designs (Fang 1980), orthogonal array-based LHD (Tang 1993; Owen 1994), orthogonal LHD (Ye 1998), uniform LHD (Jin, Chen and Sudjianto 2005), orthogonal-maximin LHD (Joseph and Hung 2008) and generalized MmLHD (Dette and Peppelyshev 2010).

All the designs discussed so far assume all the factors are quantitative. However, computer models may have both quantitative and qualitative factors (Qian, Wu and Wu 2008; Han, Santner, Notz and Bartel 2009; Hung, Joseph and Melkote 2009). Thus, we extend our notation to include qualitative variables. Let, $\boldsymbol{D} =$

$(\boldsymbol{D}_x, \boldsymbol{D}_z)$ is an n-run design, where $\boldsymbol{D}_x$ is the sub-design for p quantitative factors and $\boldsymbol{D}_z = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ is the sub-design for q qualitative factors with $l_1, \ldots, l_q$ levels and $L = \prod_{i=1}^{q} l_i$. Constructing a design with both types of factors is a challenging task. For instance, a fundamental design concept, Latin hypercube structure is very hard to apply because number of levels for a qualitative factor is usually less than n. Qian (2012) proposed sliced space filling Latin hypercube design (SLHD) for computer experiments with both quantitative and qualitative factors. An n-run p-quantitative factor SLHD is generated to have t slices each of which corresponds to a factor level combination. Each slice with $m = n/t$ runs is also an LHD. SLHD enables evaluation of computer models in batches and being LHD for each slice it can also be used for separate analysis of each batch. However, a random SLHD only has good one dimensional projections for the whole design and each slice. Ba et al. (2015) discussed that a good SLHD should be space-filling for the whole design as well as each slice. They proposed the maximin distance SLHD which is found by minimizing a weighted average of $\phi_r(\boldsymbol{D}_x)$, for the whole design, and $\phi_r(\boldsymbol{D}_{x,i})$, for each slice ($i = 1, \ldots, t$), where $\boldsymbol{D}_{x,i}$ is an m-run design corresponding to the slice i. Maximin distance SLHD is given by

$$\min_{\boldsymbol{D}_x} \quad \phi_{Mm}(\boldsymbol{D}_x) = \frac{1}{2}\left\{\phi_r(\boldsymbol{D}_x) + \frac{1}{t}\sum_{i=1}^{t}\phi_r(\boldsymbol{D}_{x,i})\right\}, \tag{35}$$

where $\boldsymbol{D}_x = ({\boldsymbol{D}_{x,1}}', \ldots, {\boldsymbol{D}_{x,t}}')'$. Criterion in (35) can also be used with other space-filling criteria, for instance, replacing MmLHD criterion with MaxPro criterion gives the MaxPro SLHD. Finally, SLHD criterion gives more emphasis to space-fillingness of the whole design than that of the slices. Maximin distance SLHD provides $\boldsymbol{D}_x$ and it ideally visions that all factor level combinations can be accommodated by slices. An $L \times m$-run SLHD is required in order to assign each level combination to a slice, however, this run size increases extremely with the number of qualitative factors. In addition, for the case, $t < L$, level combinations for each slice needs to be determined by the experimenter. Deng et al. (2015) proposed marginally coupled designs (MCDs) which can be constructed with economical run sizes. In an MCD,

$\boldsymbol{D}_z$ is an orthogonal array (OA) and $\boldsymbol{D}_x$ and corresponding design points of $\boldsymbol{D}_x$ to any qualitative factor levels are LHD . For generation of an MCD, first, an OA should exist for $\boldsymbol{D}_z$ otherwise MCD cannot be constructed. Assume that $\boldsymbol{D}_z$ is determined with an OA, next, $\boldsymbol{D}_x$ is constructed such that for each level of any qualitative factor, corresponding design points of $\boldsymbol{D}_x$ is an LHD. They discussed the existence of MCDs and proposed several construction methods for different OA structures. Although MCDs can maintain smaller design sizes, their generation has restrictions: first, an OA is required for $\boldsymbol{D}_z$, if an OA exists, second, there should be a costruction method for constructing $\boldsymbol{D}_x$. Furthermore, MCDs are only space-filling in one dimensional projections and there is no guarantee that they are space-filling in higher dimensional projections. Another space-filling design with both types of factors is Fast Flexible Filling design (Leviketz and Jones 2015) which is available in JMP 12. Leviketz and Jones (2015) used clustering-based algorithm to find the FFF designs. To generate a design with only quantitative factors, first, they randomly sample a large number of points on $\mathcal{X}$ and then select the n points with hierarchical clustering. However, we are not aware of how the design algorithm is extended to incorporate qualitative factors as well. One apparent advantage of FFF designs is their generation speed; they can be efficiently generated even for high dimensions, however, they do not possess special design properties as SLHD and MCD.

In this chapter, we propose MaxPro designs for computer experiments involving qualitative factors as well. Design criterion for MaxPro designs with quantitative factors is redeveloped to include qualitative factors. MaxPro designs generated according to the extended criterion are space-filling for all projections. A general and efficient construction method, which also uses OAs or full factorial designs, is developed.

3.2 Maximum Projection Designs with Quantitative and Qualitative Factors

In computer experiments, often, primary objective is to fit an easy-to-evaluate predictor to the output data. Gaussian process modeling, or kriging, is commonly used for modeling the predictor (Sacks, Welch, Mitchell and Wynn 1989). In the classical framework with only quantitative factors, well known ordinary kriging model is defined by,

$$Y(\boldsymbol{x}) = \mu + Z(\boldsymbol{x}), \tag{36}$$

where μ is the overall mean and $Z(\boldsymbol{x})$ is a Gaussian process (GP) with mean zero and covariance function $\sigma^2 R(\cdot)$ (Santner, Williams and Notz 2003). Gaussian correlation function is a popular choice for $R(\cdot)$ and it is given by

$$R_x(\boldsymbol{x}_i - \boldsymbol{x}_j) = \exp\{-\sum_{l=1}^{p} \alpha_l (x_{il} - x_{jl})^2\}, \tag{37}$$

where $\alpha_l \in (0, \infty)$ for $l = 1, \ldots, p$.

The maximum entropy design (Shewry and Wynn, 1987) is one of the well known design proposed for the class of Gaussian process models. Maximum entropy design is given by

$$\max_{\boldsymbol{D}_x} \det(\boldsymbol{R}_x(\boldsymbol{\alpha})), \tag{38}$$

where $\boldsymbol{R}_x(\boldsymbol{\alpha})$ is the correlation matrix with ijth element $R_x(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\alpha})$. Joseph et al.(2015) showed that MaxPro designs minimize the expected sum of off-diagonal elements of the correlation matrix for a non-informative prior distribution for $\boldsymbol{\alpha}$. They also inferred that minimizing the expected sum of off-diagonal elements tend to maximize the lower bound on the determinant of the correlation matrix. Hence, MaxPro designs are expected to perform well in terms of maximum entropy criterion which is supported by simulation results in Joseph et al. (2015). This result leads the way to the MaxPro including qualitative factors as well. To be more precise, a well-defined prior on the correlation parameters can provide a space-filling design criterion with quantitative and qualitative factors. The key to the development of this criterion is the proper definition of the correlation function. Qian, Wu and Wu (2008) proposed several correlation construction schemes along with a general framework for building Gaussian process models with both types of factors. We assume exchangeable correlation structure: correlation between two different levels of any qualitative factor is equal which is also adopted by Joseph and Delaney (2007) and Qian, Wu and Wu (2008). The correlation function for quantitative and qualitative factors is given by

$$R(\boldsymbol{w}_i - \boldsymbol{w}_j; \boldsymbol{\alpha}, \boldsymbol{\gamma}) = \exp\left\{ -\sum_{l=1}^{p} \alpha_l (x_{il} - x_{jl})^2 - \sum_{k=1}^{q} \gamma_k I[z_{ik} \neq z_{jk}] \right\}, \tag{39}$$

where $\boldsymbol{w}_i = ({\boldsymbol{x}_i}', {\boldsymbol{z}_i}')'$, $\alpha_l \geq 0$ for $l = 1, \ldots, p$, $\gamma_k > 0$ for $k = 1, \ldots, q$, and $I[z_{ik} \neq z_{jk}]$ is the indicator function that takes 1 if $z_{ik} \neq z_{jk}$ and 0 otherwise. In the context of deterministic computer experiments, response is expected to be a smooth function of the factors. Accordingly, we let each correlation parameter corresponding to the quantitative or qualitative factors has exponential distribution as priors: $\alpha_l \sim \exp(\lambda_x)$ and $\gamma_k \sim \exp(\lambda_z)$. Then, using Bayesian approach,

$$\mathbb{E}\{\sum_{i=1}^{n}\sum_{j\neq i} \boldsymbol{R}_{ij}\} = \sum_{i=1}^{n}\sum_{j\neq i} \frac{{\lambda_x}^p}{\prod_{l=1}^{p}\{(x_{il} - x_{jl})^2 + \lambda_x\}} \frac{{\lambda_z}^q}{\prod_{k=1}^{q}\{I(z_{ik} \neq z_{jk}) + \lambda_z\}}. \tag{40}$$

Thus, we extend (34) and propose MaxPro criterion in generic form,

$$\min_{\boldsymbol{D}} \quad \psi(\boldsymbol{D}) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n}\sum_{j\neq i} \frac{1}{\prod_{l=1}^{p}\{(x_{il} - x_{jl})^2 + \lambda_x\} \prod_{k=1}^{q}\{I(z_{ik} \neq z_{jk}) + \lambda_z\}}. \tag{41}$$

Proposition 1. *For an $n-$run design with p quantitative factors and one qualitative factor with t levels,*

$$\psi(\boldsymbol{D}) = \frac{1}{\lambda_z + 1}\left\{\psi(\boldsymbol{D}_x) + \frac{1}{\lambda_z}\frac{\binom{m}{2}}{\binom{n}{2}}\sum_{i=1}^{t}\psi(\boldsymbol{D}_{x,i})\right\}$$

$m = n/t$ where $\boldsymbol{D}_{x,i}$ is the sub-design of $\boldsymbol{D}_x$ corresponding to the level i, $i = 1, \ldots, t$.

Proposition 1 shows that for a design with one qualitative factor is MaxPro criterion has a similar for with SLHD criterion in (35) proposed by Ba et al. (2015). For $\lambda_z = (m-1)/(n-1)$, which is approximately equal to $1/t$, MaxPro criterion is equivalent to the SLHD MaxPro criterion. This relation shows that MaxPro design tries to achieve good projections for the whole design and also for the sub-designs corresponding to each level without forcing Latin hypercube structure.

Before the experimentation, parameters λ_x and λ_z should be specified in (41). As n distinct levels are preferred for each factor in $\boldsymbol{D}_x$, we take $\lambda_x = 0$. On the other hand, number of levels for a qualitative factor is mostly less than n, therefore, λ_z should be greater than 0, otherwise $\psi(\boldsymbol{D}) = \infty$ whenever $z_{ik} = z_{jk}$ for any k and $i \neq j$. It is hard to put a value on λ_z in advance. Thus, we adopt Bayesian framework for the selection of λ_z. By the definition in (39), correlation between z_{ik} and z_{jk}, $R(z_{ik} - z_{jk}) = \exp\{-\gamma_k I(z_{ik} \neq z_{jk})\}$ takes two values: 1 if $z_{ik} = z_{jk}$ and $\exp\{-\gamma_k\}$ otherwise. Let $\rho = \exp\{-\gamma_k\}$ and assume $\rho \backsim U(0,1)$, then by change of variables technique $\gamma_k \backsim \exp(1)$ for all $k = 1, \ldots, q$ which gives $\lambda_z = 1$. After the selection of the parameters $\lambda_x = 0$ and $\lambda_z = 1$, we propose maximum projection designs with quantitative and qualitative factors with the criterion,

$$\min_{\boldsymbol{D}} \quad \psi(\boldsymbol{D}) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j \neq i} \frac{1}{\prod_{l=1}^{p} (x_{il} - x_{jl})^2 \prod_{k=1}^{q} \{I(z_{ik} \neq z_{jk}) + 1\}}. \tag{42}$$

Before discussing the construction algorithm we would like to explore the optimality of $\boldsymbol{D}_z$. For a simple computer experiment with one qualitative factor, we find that optimal $\boldsymbol{D}_z$ is a balanced design with equal number of levels. Now, consider the problem of finding the n-run and q-qualititative factor optimal design, where $n = L$. In this case, the optimal design is the full factorial design. Design region for q qualitative factors is a regular grid in q dimensions with L nodes where each node represents a combination of levels across all factors. Since full factorial design has a design point on every node of the regular grid; it evenly fills the design space, therefore, it is space-filling. As optimal design should be balanced for each factor, OAs perform well in (42) for the problem $n < L$. This result is due to OAs' balanced structure and good projection properties in all t dimensional projections, where t is the strength of the OA. In other words, when an OA of strength t is projected onto subspace of size t, it has equal number of design points from each possible level combinations of the factors defining the subspace. Projected OA evenly covers the regular grid defined by the subspace. Thus, every t dimensional projection of an OA of strength t is space-filling. Observations on $\boldsymbol{D}_z$ are employed in finding the optimal

MaxPro designs.

3.3 Optimal Design Construction Algorithm

Finding the optimal maximum projection design is a challenging task. Since $\boldsymbol{x}$s are continuous and $\boldsymbol{z}$s are discrete, minimization of $\psi(\boldsymbol{D})$ is a mixed integer nonlinear programming problem which is very hard to solve. Moreover, number of variables $n(p+q)$ is very large. Therefore, direct optimization of $\psi(\boldsymbol{D})$ is not a viable option for finding the optimal design.

In computer experiments, a widely used optimization approach is to consider the optimal design as a Latin hypercube design and then optimize the design with exchange algorithms such as simulated annealing (Morris and Mitchell 1995) which is also used by Joseph et al.(2015) for generating the MaxPro designs with only quantitative factors. We adop the same strategy by using the findings on optimal $\boldsymbol{D}_z$ in the initialization step.

Initial design is important for the success of exchange algorithms because a good initial design mostly results in better designs in lower computation time. We initialize $\boldsymbol{D}_x$ with MaxProLHD and take $\boldsymbol{D}_z$ as an OA or a full factorial design if it exists, otherwise $\boldsymbol{D}_z$ is initialized with a balanced design. When $\boldsymbol{D}_z$ is an OA or a full factorial design, optimal design is constructed with a row-exchange algorithm which couples the rows of $\boldsymbol{D}_x$ and $\boldsymbol{D}_z$ such that $\psi(\boldsymbol{D})$ is minimized. In row-exchange algorithm, we fix $\boldsymbol{D}_z$ and in each iteration we generate a $\boldsymbol{D}_{new} = (\boldsymbol{D}_{x_{new}}, \boldsymbol{D}_z)$ where $\boldsymbol{D}_{x_{new}}$ is obtained by exchanging two randomly selected rows of $\boldsymbol{D}_x$. Next, we replace $\boldsymbol{D}$ with $\boldsymbol{D}_{new}$, if $\boldsymbol{D}_{new}$ is better than $\boldsymbol{D}$ in (42), otherwise we keep $\boldsymbol{D}$. Algorithm terminates after some stopping condition. In the second case, where there are no readily available designs for $\boldsymbol{D}_z$, we first improve a random balanced design with a fast point-exchange algorithm and use the resulting design as the initial design for $\boldsymbol{D}_z$. Point-exchange algorithm generates a new design by swapping two random elements of a design column and keeps the new design as the best design if new design improves (42) otherwise discards the new design. After initialization, $\boldsymbol{D}$ is optimized with simulated annealing algorithm. Finally, in both cases, designs are further improved to find locally optimal MaxPro-OA and MaxPro designs. These designs are obtained with an efficient derivative-based algorithm which is implemented using the gradient of the objective function,

$$\frac{\partial \psi(D)}{\partial x_{rs}} = \frac{2}{\binom{n}{2}} \sum_{i \neq r} \frac{1}{\prod_{l=1}^{p}(x_{il}-x_{rl})^2 \prod_{k=1}^{q}\{I(z_{ik} \neq z_{jk})+1\}} \frac{1}{(x_{is}-x_{rs})}. \tag{43}$$

As an example, 12-run MaxPro-OA is constructed for 2 quantitative and 2 qualitative factors each with 2 levels. In Figure 37, distribution of qualitative factors are shown on two dimensional projection of the quantitative factors. Although MaxPro designs do not necessarily posses SLHD or MCD properties, MaxPro design shown in Figure 37 is an SLHD.

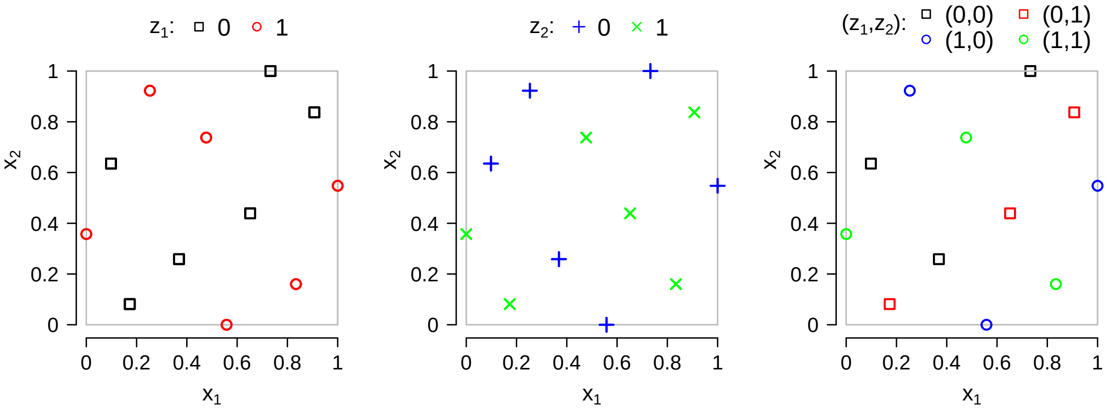

Figure 37: Projections of a 12-run MaxPro-OA

3.4 Numerical Results

In this section, we compare the performance of MaxPro designs with MCD, SLHD and fast flexible filling (FFF) designs for two different cases:

Case 1: $n = 25$, $p = 3$ quantitative and $q = 3$ qualitative factors each with 5 levels

Case 2: $n = 100$, $p = 5$ quantitative and $q = 5$ qualitative factors each with 5 levels

In case 1, an OA$(25, 5^6)$ with 25 runs and six 5-level factors is used for constructing an MCD. First three columns are used for qualitative factors and the last three columns are transformed to a random LHD using the construction procedure for s-level OAs with s^2 runs proposed in Deng et al.(2015). However, in case 2, MCD is not included in comparison; it cannot be constructed because there is no available OA. FFF designs are constructed on JMP with 1000 random starts for both cases. Finally, SLHDs are constructed using the R package SLHD (Ba 2015) with 1000 random starts. Then, we determine the level combinations for each slice as follows. First, for both cases, we consider each slice has 5 runs so that each slice has enough runs to achieve space-fillingness. Thus, there are 5 and 20 slices for cases 1 and 2, respectively. Originally, there is no defined scheme on selection of level combinations for each slice. We determine the level combinations using maximin distance criterion. For case 1, we optimize a 5-run 3-qualitative factor design by maximizing the minimum Hamming distance between any two level combinations. For $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, Hamming distance is calculated with $\tilde{d}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sum_{k=1}^{q} I(z_{ik} \neq z_{jk})$. In the last step, SLHDs are constructed by randomly assigning level combinations to each slice. For generation of MaxPro designs, in both cases, $\boldsymbol{D}_x$ is initialized with MaxProLHDs generated by the R Package MaxPro (Ba and Joseph 2015). On the other hand, for the first case, $\boldsymbol{D}_z$ is initialized with the same OA which is used for MCD. For the second case, as there is no available OA, $\boldsymbol{D}_z$ is initialized with an improved balanced design. Next, initial designs are optimizated with row-exhange algorithm for the first case and simulated annealing algorithm for the second case. Finally, MaxPro-OA and MaxPro designs are obtained with continuous optimization algorithm in the last step of the construction algorithm.

3.4.1 Numerical Results for Case 1

Case 1 designs are first compared in two space-filling criteria: maximin and minimax distance. In order to incorporate index of the design, we use Mm_k, maximin, and mM_k, minimax distance criteria defined by Joseph et al. (2015). For a design with only quantitative factors, these measures are calculated for each projection of a given subdimension k, $k = 1 \dots, p$. Then the worst case of all possible k-dimensional projections is compared. Mm_k is given by

$$Mm_k = \min_{r=1,\dots,\binom{p}{k}} \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d_{kr}^{2k}(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right\}^{-1/(2k)},$$

where $d_{kr}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Euclidean distance between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the rth projection of dimension k. Since distance definitions of quantitative and qualitative factors are not compatible, Mm_k is separately defined and reported for quantitative and qualitative parts of the design. Then, $\widetilde{Mm}_k$ is defined as,

$$\widetilde{Mm}_k = \min_{r=1,\dots,\binom{q}{k}} \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\tilde{d}_{kr}^{2k}(\boldsymbol{z}_i, \boldsymbol{z}_j) + 1} \right\}^{-1/(2k)},$$

where $\tilde{d}_{kr}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is the Hamming distance between points $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ in the rth projection of dimension k, $k = 1, \dots, q$. There may be some replications in some projections, therefore, we add one to the denominator of $\widetilde{Mm}_k$ to prevent it to become infinity. Figure 38 shows the Mm_k and $\widetilde{Mm}_k$ measures. MaxPro-OA performs the best in Mm_k. Having OA for qualitative part of their designs, MaxPro-OA and MCD perform equally well and they are significantly better than SLHD and FFF designs in $\widetilde{Mm}_k$. Although MCD is not optimized with a space-filling criteria, it does better than FFF in both criteria. SLHD also performs better FFF in Mm_k, however, it performs significantly worse in $\widetilde{Mm}_k$ because it has many replications in each projection which significantly deteriorates the space-fillingness. Next, the minimax distance measure, mM_k is given by

$$mM_k = \max_{r=1,\dots,\binom{p}{k}} \max_{\boldsymbol{x} \in \mathcal{X}_k} \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{d_{kr}^{2k}(\boldsymbol{x}, \boldsymbol{x}_i)} \right\}^{-1/(2k)},$$

where $\mathcal{X}_k$ is the set of sample points used to find the approximate maximum weighted distance between the design points and the design region. It consists of 3^k factorial design with levels $\{0, 0.5, 1\}$ and 2^{16}-run Sobol sequence. Similarly, mM_k measure is

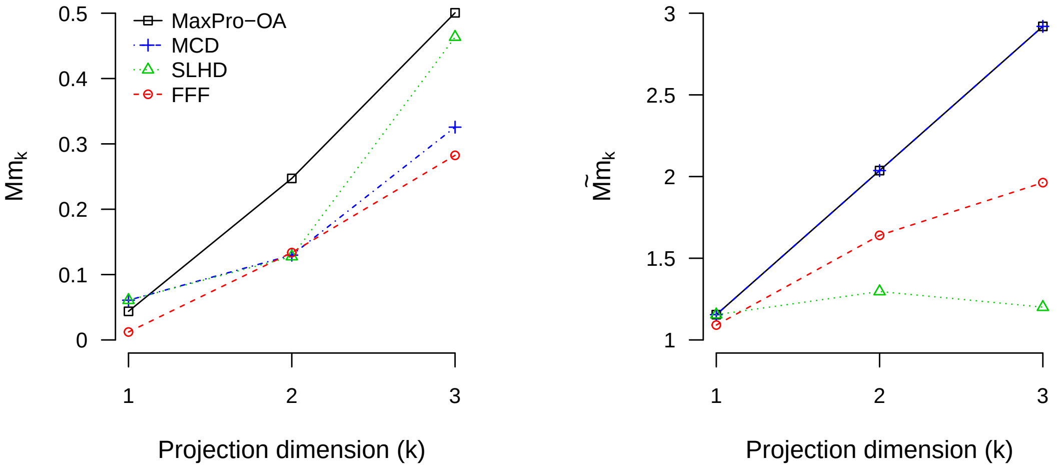

Figure 38: Mm_k and $\tilde{Mm}_k$ for case 1 designs (larger-the-better)

modified for qualitative factors and $\tilde{mM}_k$ is defined as

$$\tilde{mM}_k = \max_{r=1,\ldots,\binom{p}{k}} \max_{\boldsymbol{z}\in\mathcal{Z}_{kr}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\tilde{d}_{kr}^{2k}(\boldsymbol{z},\boldsymbol{z}_i)+1} \right\}^{-1/(2k)},$$

where $\mathcal{Z}_{kr}$ is a full factorial design of the factors defined by the r^{th} projection of subdimension k. In case 1, $\mathcal{Z}_{kr}$ is a 5^k factorial design. These two measures plotted in Figure 39. In both mM_k and $\tilde{mM}_k$, MaxPro-OA is superior to other designs. FFF is the worst in mM_k because it has big gaps in the design region. On the other hand, MCD has comparable performance to SLHD and it has the best performance in $\tilde{mM}_k$ as MaxPro-OA.

After comparing designs in space-filling criteria, we further assess the performance of the designs in Gaussian process modeling criteria. As discussed earlier maximum entropy design maximizes the determinant $|R(\boldsymbol{\alpha},\boldsymbol{\gamma})|$, where $R(\boldsymbol{\alpha},\boldsymbol{\gamma})$ is the correlation matrix with (i,j) element is $R(\boldsymbol{w}_i-\boldsymbol{w}_j;\boldsymbol{\alpha},\boldsymbol{\gamma})$. For a subdimension k, designs are compared in minimum correlation matrix determinant, $\min_r \log|R^{k,r}(\boldsymbol{\alpha},\boldsymbol{\gamma})|$, where $R^{k,r}$ is the r^{th} projection. In order to compare designs in maximum entropy criterion we need to set the correlation parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$. The mean of prior for qualitative

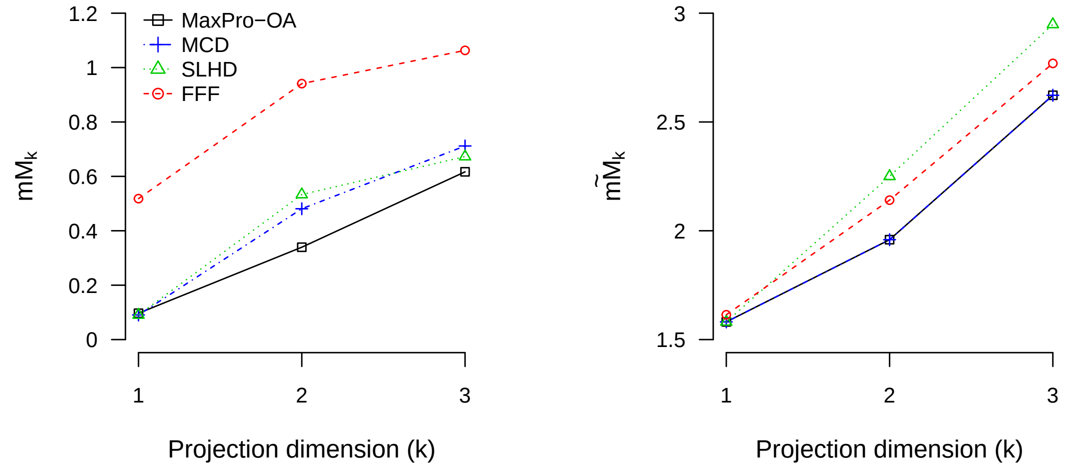

Figure 39: mM_k and $m\tilde{M}_k$ for case 1 designs (smaller-the-better)

factors being equal to 1, as a reasonable choice, correlation parameter γ_l is set to 1 for $l = 1, \ldots, q$. In all reason, first choice for α_l should also be 1 for $l = 1, \ldots, p$, however, this creates a large discrepancy between the distance values of qualitative factors and quantitative factors in correlation function (39). Consequently, assuming equal values α_l and γ_l gives more importance to qualitative factors. In order to balance the importance of the factors, we take $\alpha_l = 3$ and $\alpha_l = 4$ for the first and second cases, respectively. Minimum log-determinants of the designs are plotted in 40. As seen in Figure 40, MaxPro-OA outperforms other designs. MCD performs better than SLHD and FFF designs. SLHD performs bad in projection dimensions 2 and 3 since in these dimensions projections with only qualitative factors have significantly small determinant values because of the replications.

Ordinary kriging model for a computer experiment with both types of factors is given by,

$$Y(\boldsymbol{w}) = \mu(\boldsymbol{\alpha}, \boldsymbol{\gamma}) + Z(\boldsymbol{w}), \tag{44}$$

where $\boldsymbol{w}_i = (\boldsymbol{x}_i{}', \boldsymbol{z}_i{}')'$, μ is the overall mean and $Z(\boldsymbol{w})$ is a GP with mean zero and covariance function $\sigma^2 R(\cdot)$. The ordinary kriging predictior is

$$\hat{y}(\boldsymbol{w}) = \hat{\mu}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) + \boldsymbol{r}(\boldsymbol{w}; \boldsymbol{\alpha}, \boldsymbol{\gamma})' \boldsymbol{R}^{-1}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) \{\boldsymbol{y} - \hat{\mu}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) \boldsymbol{1}_n\},$$

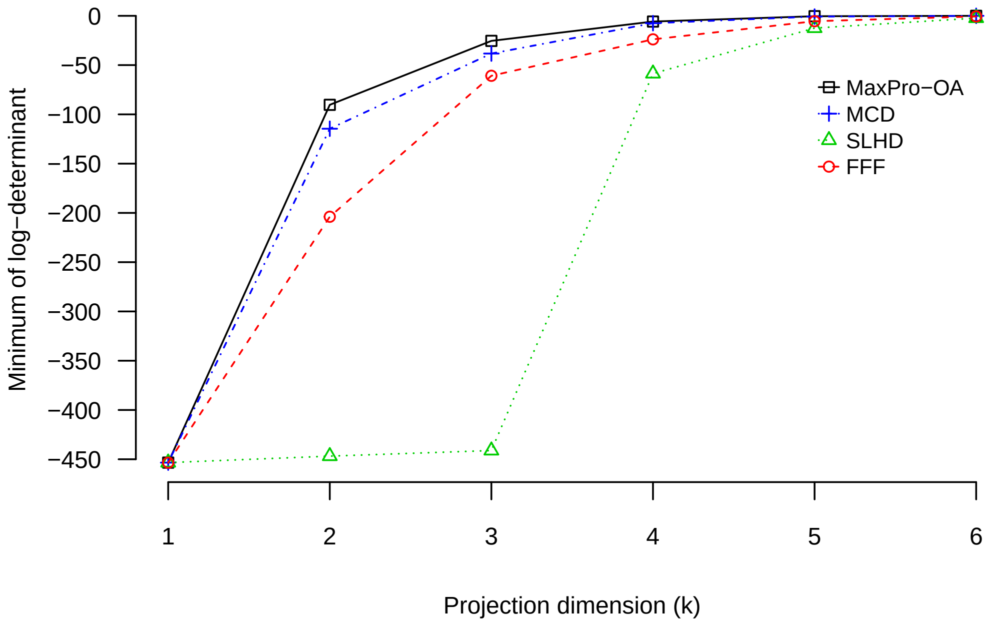

Figure 40: Minimum log-determinant for case 1 designs (larger-the-better)

where $\hat{\mu}(\boldsymbol{\alpha},\boldsymbol{\gamma}) = \mathbf{1}_n'\boldsymbol{R}^{-1}(\boldsymbol{\alpha},\boldsymbol{\gamma})\boldsymbol{y}/\mathbf{1}_n'\boldsymbol{R}^{-1}(\boldsymbol{\alpha},\boldsymbol{\gamma})\mathbf{1}_n$, $\boldsymbol{r}(\boldsymbol{w};\boldsymbol{\alpha},\boldsymbol{\gamma})$ is a vector of length n with i^{th} element $R(\boldsymbol{w}-\boldsymbol{w}_i;\boldsymbol{\alpha},\boldsymbol{\gamma})$, $\boldsymbol{y} = (y_1,\ldots,y_n)'$ is the experimental data, and $\mathbf{1}_n$ is a vector of 1's having length n. Prediction is done at the optimal values of $(\boldsymbol{\alpha},\boldsymbol{\gamma})$ which are found by minimizing the negative log-likelihood (Santner, Williams and Notz 2003 pp. 66),

$$\log|\boldsymbol{R}(\boldsymbol{\alpha},\boldsymbol{\gamma})| + n\log\hat{\sigma}^2(\boldsymbol{\alpha},\boldsymbol{\gamma})$$

with respect to $(\boldsymbol{\alpha},\boldsymbol{\gamma})$, where $\hat{\sigma}^2(\boldsymbol{\alpha},\boldsymbol{\gamma}) = \{\boldsymbol{y}-\hat{\mu}(\boldsymbol{\alpha},\boldsymbol{\gamma})\mathbf{1}_n\}'\boldsymbol{R}^{-1}(\boldsymbol{\alpha},\boldsymbol{\gamma})\{\boldsymbol{y}-\hat{\mu}(\boldsymbol{\alpha},\boldsymbol{\gamma})\mathbf{1}_n\}/n$. Usually, an iterative optimization algorithm is used and it requires the evaluation of $\boldsymbol{R}^{-1}(\boldsymbol{\alpha},\boldsymbol{\gamma})$ for hundreds of times. For some values of $(\boldsymbol{\alpha},\boldsymbol{\gamma})$, computation of $\boldsymbol{R}(\boldsymbol{\alpha},\boldsymbol{\gamma})$ can be cumbersome and unstable, therefore, a design which avoids this problem for all values of $(\boldsymbol{\alpha},\boldsymbol{\gamma})$ is preferable. Condition number of a matrix is a commonly used criterion for assessing the instability of its inverse. Maximum condition numbers for each projection dimension are plotted in Figure 41. MaxPro-OA design is the best in all projection dimensions. SLHD and FFF designs are the worst performers in this criterion.

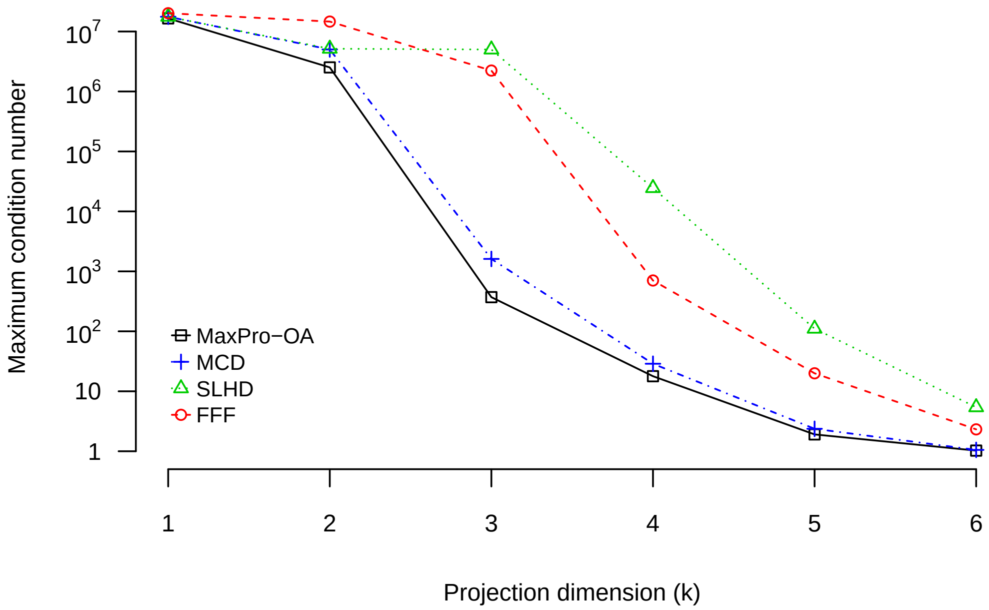

Figure 41: Maximum condition number for case 1 designs (smaller-the-better)

Our last Gaussian process modeling criterion is the maximum prediction variance in the design space. The prediction variance for ordinary kriging is proportional to,

$$1 - \boldsymbol{r}(\boldsymbol{w};\boldsymbol{\alpha},\boldsymbol{\gamma})'\boldsymbol{R}^{-1}(\boldsymbol{\alpha},\boldsymbol{\gamma})\boldsymbol{r}(\boldsymbol{w};\boldsymbol{\alpha},\boldsymbol{\gamma}) + \frac{\{1 - \boldsymbol{r}(\boldsymbol{w};\boldsymbol{\alpha},\boldsymbol{\gamma})'\boldsymbol{R}^{-1}(\boldsymbol{\alpha},\boldsymbol{\gamma})\mathbf{1}_n\}^2}{\mathbf{1}_n'\boldsymbol{R}^{-1}(\boldsymbol{\alpha},\boldsymbol{\gamma})\mathbf{1}_n}.$$

For a given projection, the maximum prediction variance is approximated by merging the set of points which are used for approximation of minimax measure for quantitative and qualitative parts of the designs. For a given subdimension k, 5^k full factorial design for qualitative factors is replicated to match the size of set of points for quantitative factors. Maximum prediction variances are plotted in Figure 42 where MaxPro-OA is the winner.

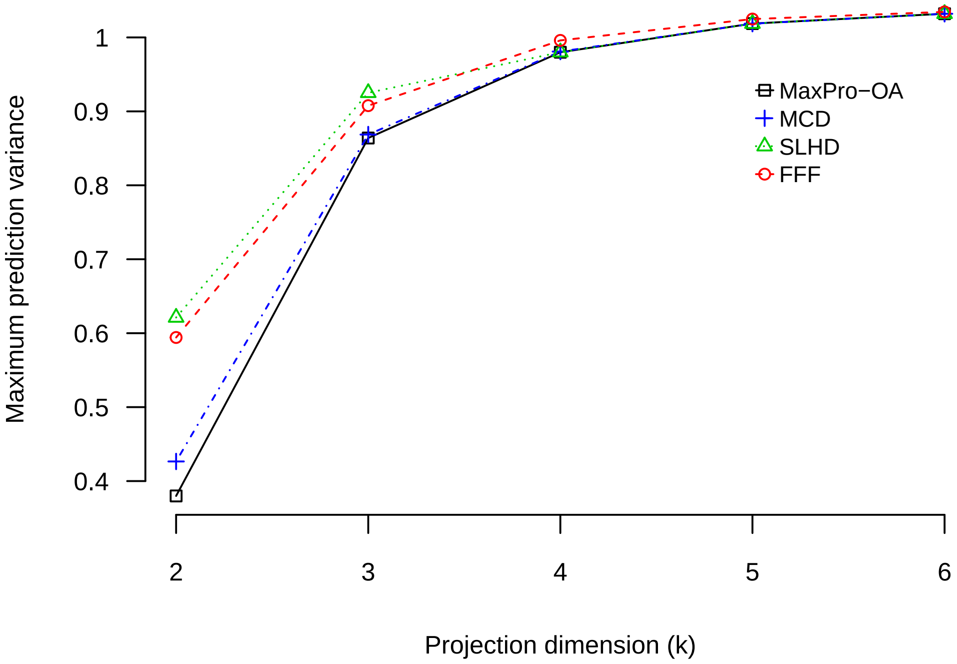

Figure 42: Maximum prediction variance for case 1 designs (smaller-the-better)

3.4.2 Numerical Results for Case 2

In this section, MaxPro design, SLHD and FFF design are compared in a bigger design setting: 100-runs with 5 quantitative and qualitative factors where each qualitative factor has 5 levels. First, Figure 43 shows the Mm_k and $\widetilde{Mm}_k$ measures for the designs. SLHD has the highest Mm_k for the full design since it behaves like a maximin Latin hypercube design, whereas, MaxPro design is better than SLHD and FFF design in lower dimensional projections. When we compare designs in $\widetilde{Mm}_k$, we see that MaxPro design has the best performance. On the other hand, performances of SLHD and FFF design deteriorate as projection dimension increases. Next, designs are compared in minimax measures in Figure 44. MaxPro design and SLHD have comparable performance in mM_k, whereas, FFF design is significantly worse than others. On the other hand, MaxPro design is uniformly better than SLHD and FFF design in $\widetilde{mM}_k$.

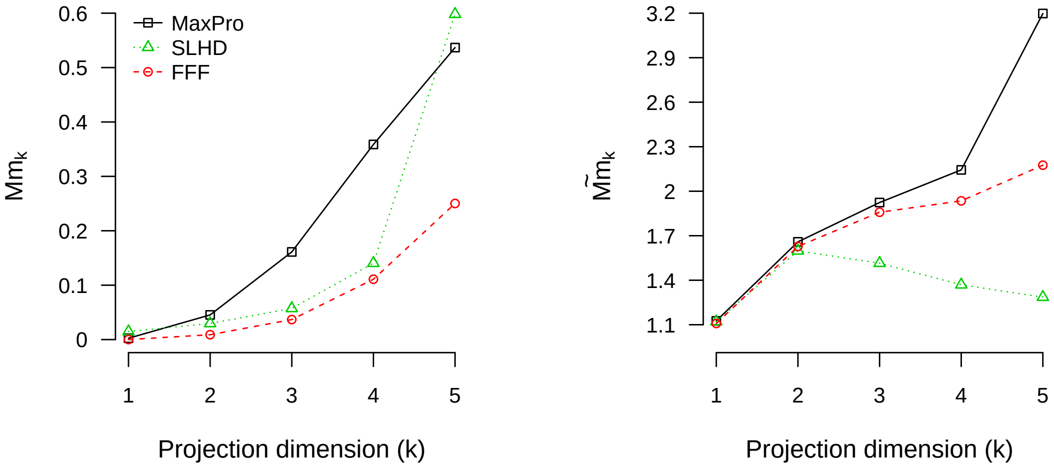

Figure 43: Mm_k and $\tilde{M}m_k$ for case 2 designs (larger-the-better)

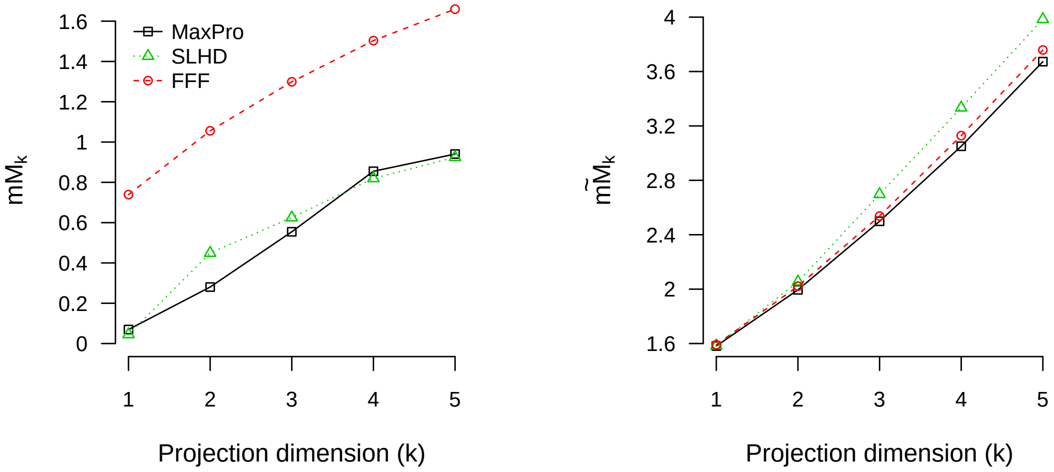

Figure 44: mM_k and $m\tilde{M}_k$ for case 2 designs (smaller-the-better)

Now, case 2 designs are compared in Gaussian process modeling criteria. Maximum entropy measure is plotted in Figure 45. As in case 1, MaxPro design is the winner in maximum entropy criterion. Similar to case 1, SLHD underperforms in lower dimensional projections from 2 to 5 due to the replications in the qualitative part of the design. Figure 46 shows the maximum condition numbers for all subdimensions. MaxPro design has the smallest maximum condition numbers for every

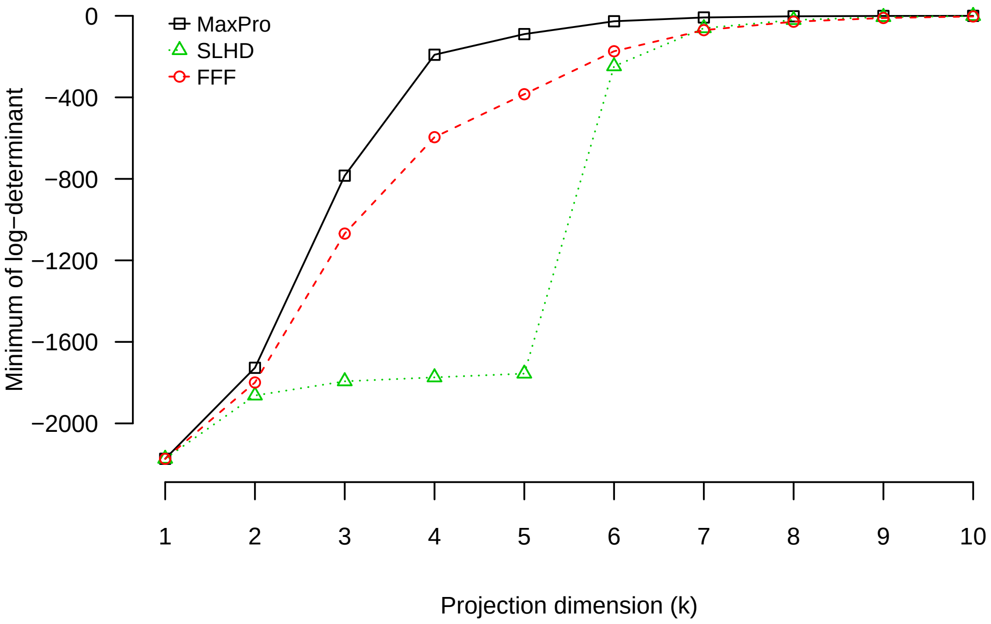

Figure 45: Minimum log-determinant for case 2 designs (larger-the-better)

projection dimension. Finally, maximum prediction variances are plotted in Figure 47 where MaxPro design is superior to SLHD and FFF design.

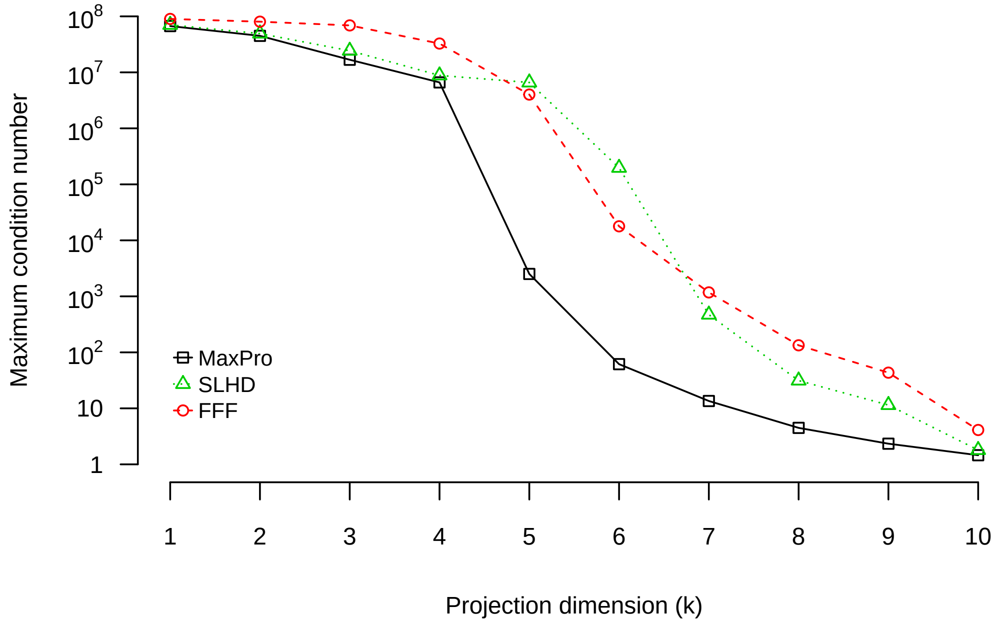

Figure 46: Maximum condition number for case 2 designs (smaller-the-better)

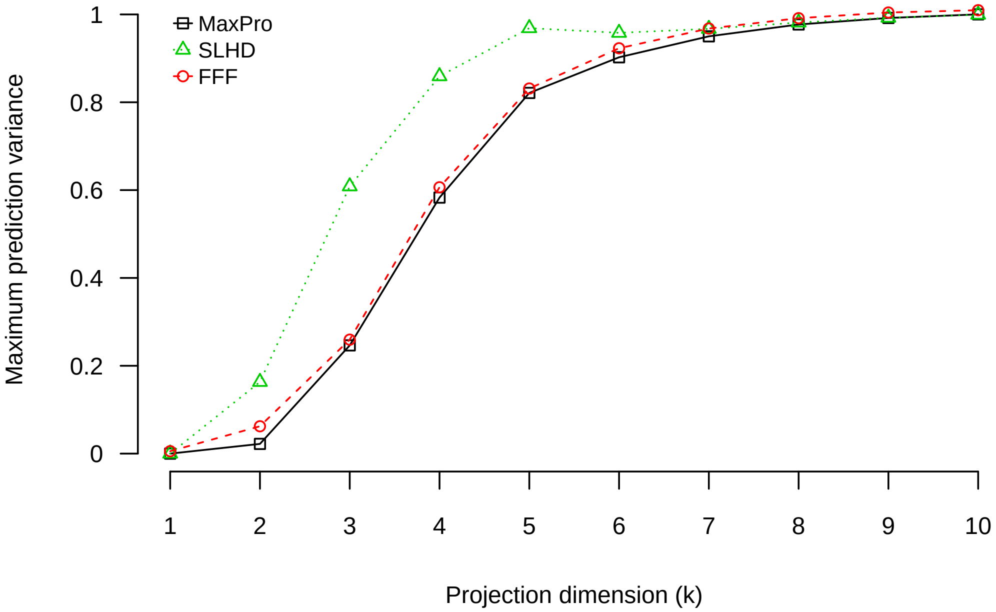

Figure 47: Maximum prediction variance for case 2 designs (smaller-the-better)

3.5 Example: Solid End Milling Process

We illustrate the performance of MaxPro designs using a simulation conducted for optimization of a 5-axis solid end milling process. End milling is a widely used cutting process in aerospace industry. Precise and fast end milling of aerospace structure components is of great importance. We will optimize a solid end milling process simulation of a component belonging to an aerospace structure. Six factors are selected for the optimization. Four of the factors are quantitative and they are all related to the cutting tool: rake angle (x_1), helix angle (x_2), relief angle (x_3) and corner radius (x_4). Cutting tool parameters are shown in Figure 48. Two qualitative factors are titanium alloy (z_1) and tool path optimization type (z_2). Six different titanium alloys, which are commonly used in aerospace in industry (Donachie 2000, 2^{nd} Chapter), and four available optimization types are considered for the optimization. Qualitative factors and their levels are given in Table 4. The simulation is done on the Production Module of Third Wave Systems (Minneapolis, MN). One output of the end milling process simulation and the workpiece is shown in Figure 49.

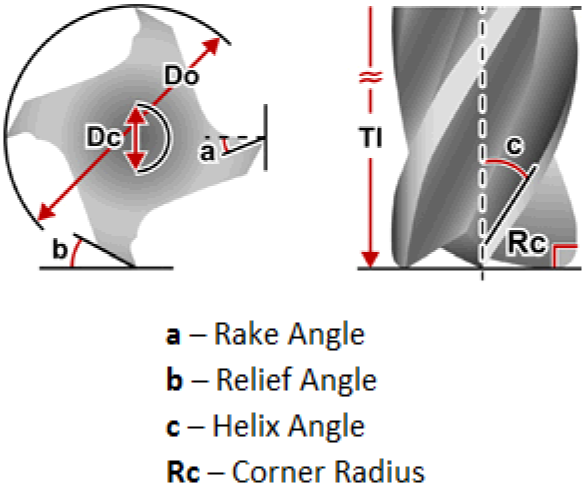

Figure 48: End milling cutting tool parameters

In cutting processes, a commonly used objective in process optimization is to maximize the tool life and prevent the tool breakages by lowering the maximum tangential

Table 4: Qualitative Factors and Levels

Level	Titanium Alloy	Tool Path Optimization
0	Ti-6AI-4V	None
1	Ti-6AI-2Sn-4Zr-6Mo	In-Cut
2	Ti-6AI-2Sn-4Zr-2Mo	Air-Cut
3	Ti-6AI-6V-2Sn	Both
4	Ti-4AI-4Mo-2Sn	
5	Ti-10V-2Fe-3AI	

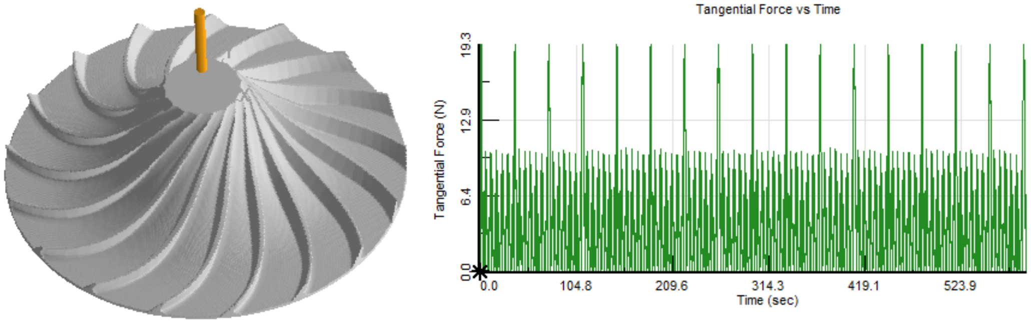

Figure 49: Solid end milling process simulation on the Production Module: Workpiece and the tangential force output

force. Thus, the objective of the experiment is to find factor settings which minimizes the maximum tangential force. Ranges for quantitative factors are selected as follows: $x_1 \in [3.5, 6.5]$ deg, $x_2 \in [21, 39]$ deg, $x_3 \in [7, 13]$ deg, $x_4 \in [0.063, 0.117]$ mm. A 24-run MaxPro-OA is generated where $\boldsymbol{D}_z$ is a full factorial design. We fit an ordinary kriging model to the data. The correlation parameters are estimated efficiently as MaxPro design provides well-conditioned correlation matrix. For this simulation data, correlation parameters are estimated as $\hat{\boldsymbol{\alpha}} = (0.009, 1.588, 0.001, 0.068)'$ and $\hat{\gamma} = (0.724, 0.029)'$. In order to provide an insight about the model, main effects for six factors are derived using functional analysis of variance decomposition. Main effects of the four quantitative factors are given in Figure 50 and two qualitative factors are given in Figure 51. We see that helix angle (x_2) and titanium alloy type (z_1) have the largest effects. Next, optimal factor settings minimizing the maximum tangential

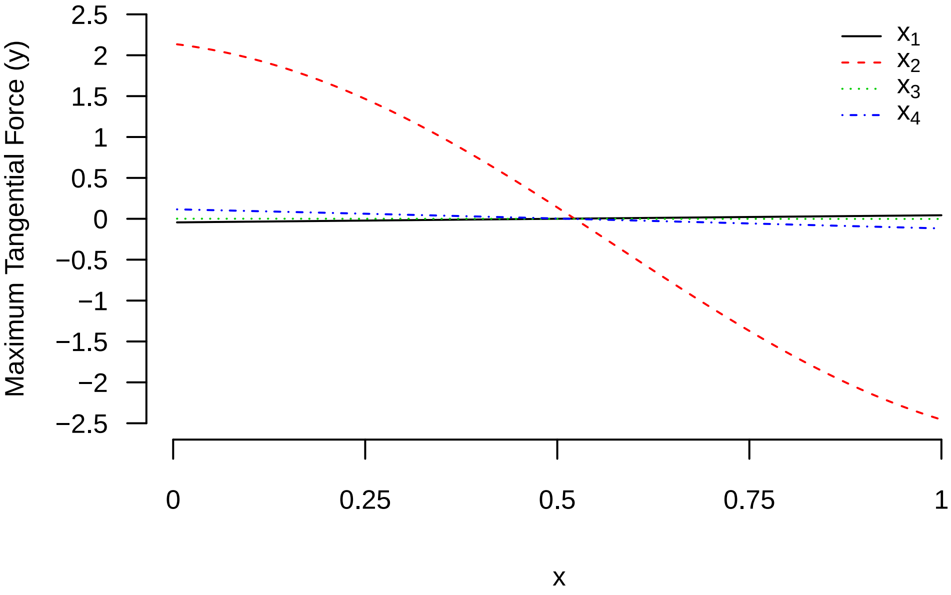

Figure 50: Main effects of the four quantitative factors

force are found as $x_1^* = 0$, $x_2^* = 1$, $x_3^* = 1$, $x_4^* = 1$, $z_1^* = 0$, and $z_2^* = 1$. The predicted maximum tangential force at the optimal setting is 11.57 N and the actual maximum tangential force is 11.45 N. The model is also validated by running Production Module at 40 settings. Fixing $z_1 = 0$ and $z_2 = 1$, quantitative factor settings are determined with a 40-run uniform design generated using JMP software. Coefficient of variation of the root mean square error, (RMSE/$\bar{y}$), where $\bar{y}$ is the mean of the maximum tangential forces, is calculated as 0.47%. Therefore, the ordinary kriging predictor is a very accurate metamodel for the actual computer model. Another alternative design for modeling is the SLHD. However, a large SLHD needs to be generated since there are 24 slices corresponding to the total level combinations of the two qualitative factors. If we reasonably consider 10 runs for each slice, then it forms a 240-run SLHD which is hard to generate and model. On the other hand, MaxPro design with much less number of runs ensures accurate and efficient modeling.

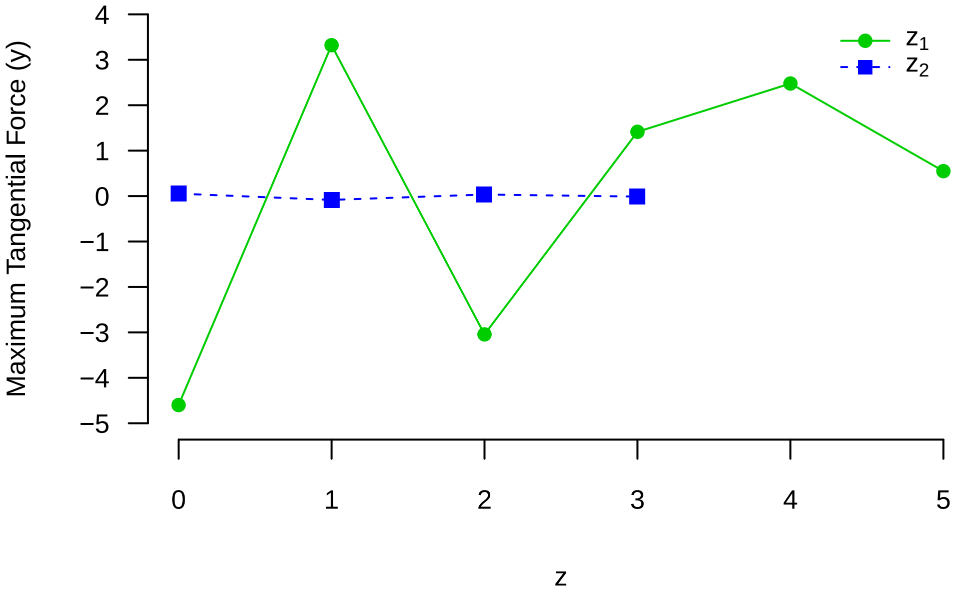

Figure 51: Main effects of the two qualitative factors

3.6 Sequential Construction of Maximum Projection Designs

In many industries, improvement of a product/process with lower financial cost and in minimum time is crucial to remain competitive. Thus, design and modeling of computer experiments should be efficient. In general, computer experiments are modeled with a nominal design size. A common approach in practice is to take $n = 10(p + q)$. However, this one-shot-approach has two main drawbacks. First, the design can be large such that prediction accuracy of the model is over the requirements which means waste of resources and unnecessary increase in cost. Second, the design can be small to achieve a model with desired prediction capability. A solution for these problems is to construct designs sequentially and stop at the design which provides a model with desired prediction accuracy.

Maximum projection designs do not have any restrictions in their construction, therefore, they can be constructed sequentially which is not possible for SLHDs or

MCDs. In sequential construction of designs, a commonly used approach is to start with an initial design or a point and add points one-at-a-time or in batches according to a criterion. Some of the proposed sequential designs may also have desired properties. Recently, Xu et al. (2015) proposed a sequential construction approach which adds design points in batches and preserves Latin hypercube structure at any stage of the construction. Similarly, Duan et al.(2015) introduced an batch sequential design algorithm using sliced full factorial-based Latin hypercube designs which is also an LHD in every step of the design and has good projection properties at some stages.

In general, next design points are selected adaptively using the data generated until that time. However, in our approach we add the next point based on a distance-based criterion. We start with an initial point and add points one-at-a-time according to the criterion. Candidate sets are generally employed for selecting the next point to add to the design. Usually, a large ensemble (size > 10000) of quasi-random sequences is used as a candidate set. However, using large size of quasi-random sequences can be time-consuming and may not quarantee a competent design for maximum affordable design size N which is dictated by our budget. An intelligent alternative is to use an N-run optimal design as the candidate set. By doing so sequential construction is faster and guarantees that final design is an optimal design. Let $\boldsymbol{D} = \{({\boldsymbol{x}_1}', {\boldsymbol{z}_1}')', \ldots ({\boldsymbol{x}_N}', {\boldsymbol{z}_N}')'\}$ be an N-run $(p+q)$-factor MaxPro design. Points of $\boldsymbol{D}$ constitutes a candidate set of size N which is large enough to obtain good designs through the sequential construction. Starting with a random point in $\boldsymbol{D}$, we add $N-1$ points one-at-a-time. Suppose that we have generated $({\boldsymbol{x}_1}', {\boldsymbol{z}_1}')', \ldots ({\boldsymbol{x}_n}', {\boldsymbol{z}_n}')'$, then we can find $({\boldsymbol{x}_{n+1}}', {\boldsymbol{z}_{n+1}}')'$ as

$$({\boldsymbol{x}_{n+1}}', {\boldsymbol{z}_{n+1}}')' = \underset{(\boldsymbol{x}', \boldsymbol{z}')' \in \boldsymbol{D}}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{1}{\prod_{l=1}^{p} (x_l - x_{il})^2 \prod_{k=1}^{q} \{I(z_k \neq z_{ik}) + 1\}}. \tag{45}$$

After adding $N-1$ points to a random point, we obtain an N-run design whose first n runs is the n-run sequential design where $n = 1, \ldots, N$. Although each starting point reaches optimal design at the end of the algorithm, some of them may generate bad designs for intermediate design sizes. In practice, N experiments can be afforded, however, it is desired to choose the starting point which achieves the model requirements with the smallest design size. A naive approach for finding the best starting point is to find the smallest design size for each starting point and then get the best starting point as the one with minimum smallest design size. However, this approach requires fitting many models, and so, it can be very computationally intensive. As an alternative approach, we can use a nominal design size, $\bar{n}$, and choose the starting point with the best $\bar{n}$-run sequential design. Then, the best starting point can be found as

$$(\boldsymbol{x}_*^{'}, \boldsymbol{z}_*^{'})^{'} = \underset{(\boldsymbol{x}^{'},\boldsymbol{z}^{'})^{'}\in\boldsymbol{D}}{\operatorname{argmin}} \psi(\boldsymbol{D}_{\bar{n}}(\boldsymbol{x}, \boldsymbol{z})), \tag{46}$$

where $\boldsymbol{D}_{\bar{n}}(\boldsymbol{x}, \boldsymbol{z})$ is the $\bar{n}$-run sequential design generated with the starting point $(\boldsymbol{x}^{'}, \boldsymbol{z}^{'})^{'}$. Then $\boldsymbol{D}_N^*$ and $\boldsymbol{D}_{\bar{n}}^*$ are the best sequential N-run and $\bar{n}$-run designs generated with $(\boldsymbol{x}_*^{'}, \boldsymbol{z}_*^{'})^{'}$.

Optimality criterion (46) ensures that we obtain a $\bar{n}$-run sequential closest to the optimal $\bar{n}$-run MaxPro design. Design size, $\bar{n}$ can be considered as the size of the nominal design use we use in one shot approach, thus, $\bar{n} = 10(p + q)$ can be a reasonable choice. We expect a well-constructed $\bar{n}$-run design will generate good designs until the end of the sequential construction. Therefore, $\bar{n}$ should be large enough to reach a well-located and robust design so that it will perform good in the rest of the construction and it should be small enough to not to miss good designs for moderate design sizes. Proposed methodology is demonstrated on two examples.

3.6.1 Example 1

Suppose that we want to construct a sequential design with $p = 2$ quantitative factors and $q = 1$ qualitative factor with 3 levels. An $N = 90$-run optimal MaxPro-OA design is generated as the candidate set and $\bar{n}$ is taken as 30. Sequential designs are constructed with (45) and the best sequential design is determined using (46). Figure 52 shows the MaxPro criterion values of the optimal $\bar{n}$ and N-run MaxPro designs andthe best sequential design and the range of MaxPro criterion values of all designs throughout the sequential construction. As seen in Figure 52, after $\bar{n}$ best sequential design continues to have small criterion values until N.

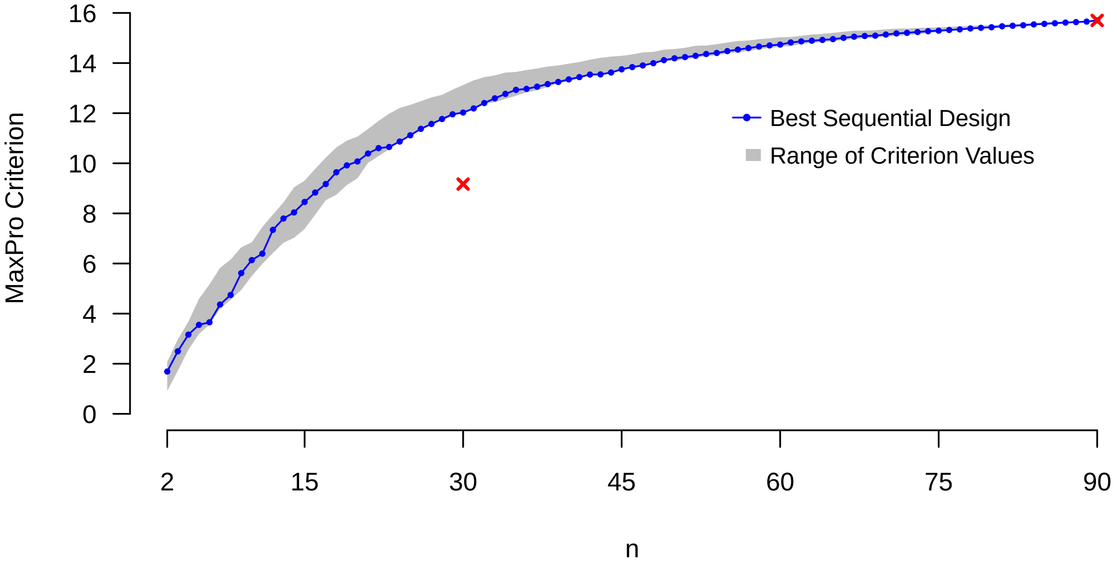

Figure 52: Example 1: Best sequential design and range of MaxPro criterion values throughout the sequential construction, red crosses show the MaxPro criterion values for the $\bar{n}$ and N-run MaxPro designs

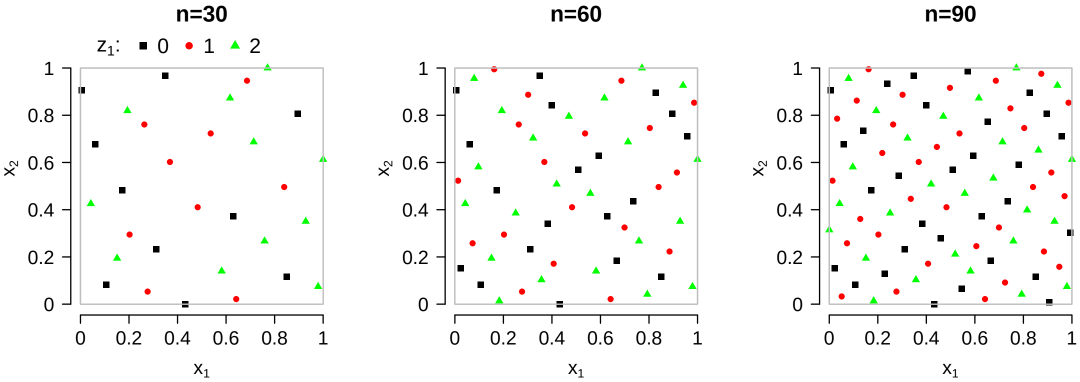

Figure 53: Two dimensional projections of the best design for $n = 30, 60$ and 90

Two dimensional projections of the best design onto the quantitative factors are plotted in Figure 53 for n = 30, 60 and 90. Projections seem space-filling for each level of the qualitative factors in Figure 53. In order to understand space-fillingness of sub-designs corresponding to each level, we plot the minimum distances of the sub-designs for $n = 2, \ldots, N$ in Figure 54. It is seen in Figure 54 that none of the levels are dominating and they are similarly space-filling after $n = 20$ with respect to the minimum distance criterion. In the next example, we apply sequential construction methodology on the optimization of solid end milling process studied in section 3.5.

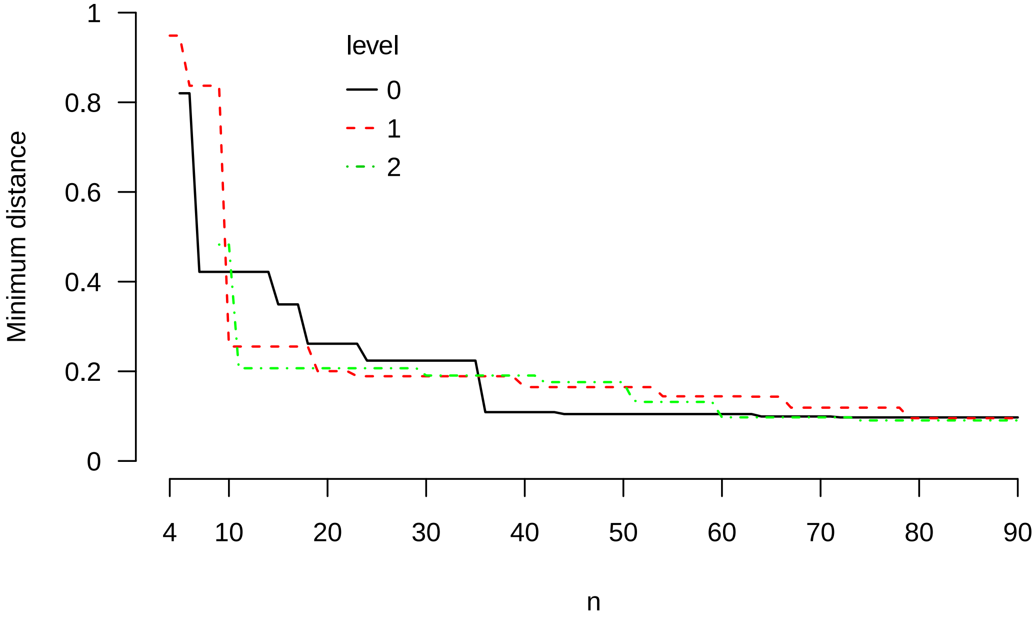

Figure 54: Minimum distances of the design points corresponding to each level of the qualitative factor

3.6.2 Example 2

Although we modeled solid end milling process with a 24-run MaxPro-OA in section 3.5, our budget allows us to conduct 60 simulations. Thus, sequential construction methodology is performed with $N = 60$ and $\bar{n} = 24$. MaxPro criterion values for the best design and minimum and maximum criterion values for $n = 2, \ldots, 60$ are plotted in Figure 55. Again, the best starting point yields good designs at $\bar{n} = 24$ and forward in MaxPro criterion. In optimization of the solid end milling process, we want to build a competent kriging model with the lowest number of simulations. Therefore, we sequentially fit ordinary kriging models and obtain the best model as the first model with $RMSECV/\bar{y}$ value smaller than or equal to 1%, where $RMSCEV$ is root mean square error of cross validation. Figure 56 shows the percentage $RMSECV/\bar{y}$ for the models built using sequential designs with $n = 12$ to 24. According to Figure 56, the first model which meets the prediction criterion is achieved at $n = 20$. Thus, using sequential design methodology we obtain a good model with lesser number of runs than the 24-run design we used in section 3.5. As a result, using sequential MaxPro designs computer simulations can be modeled with an economical run size. Proposed methodology can be further improved by incorporating the data in the design criterion which enables efficient exploration of the response surface. We will study this extension of the sequential design criterion in a future work.

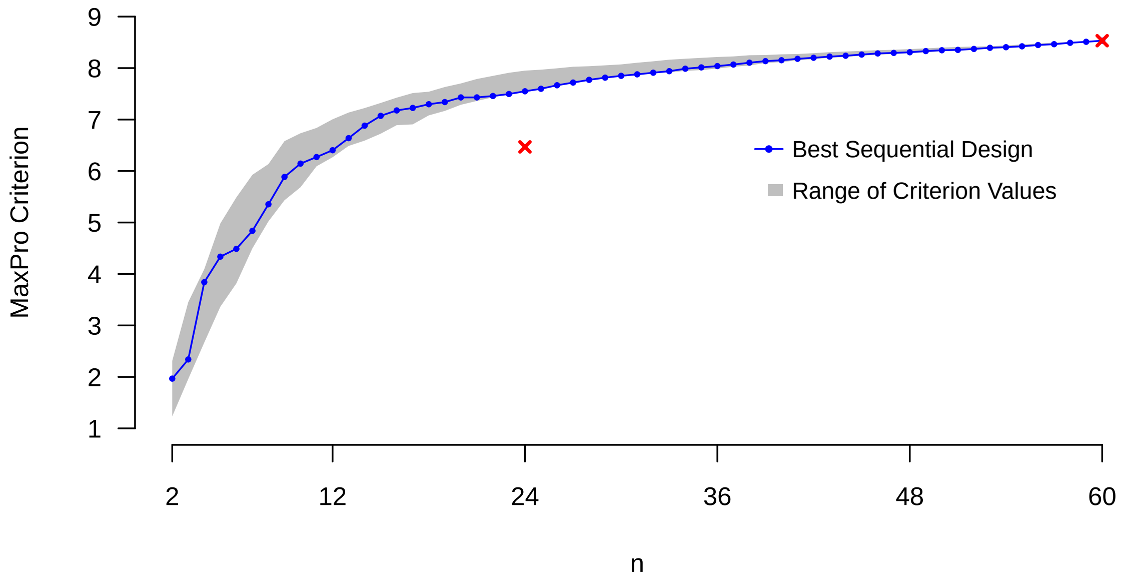

Figure 55: Example 2: Best sequential design and range of MaxPro criterion values throughout the sequential construction, red crosses show the MaxPro criterion values for the $\bar{n}$ and N-run MaxPro designs

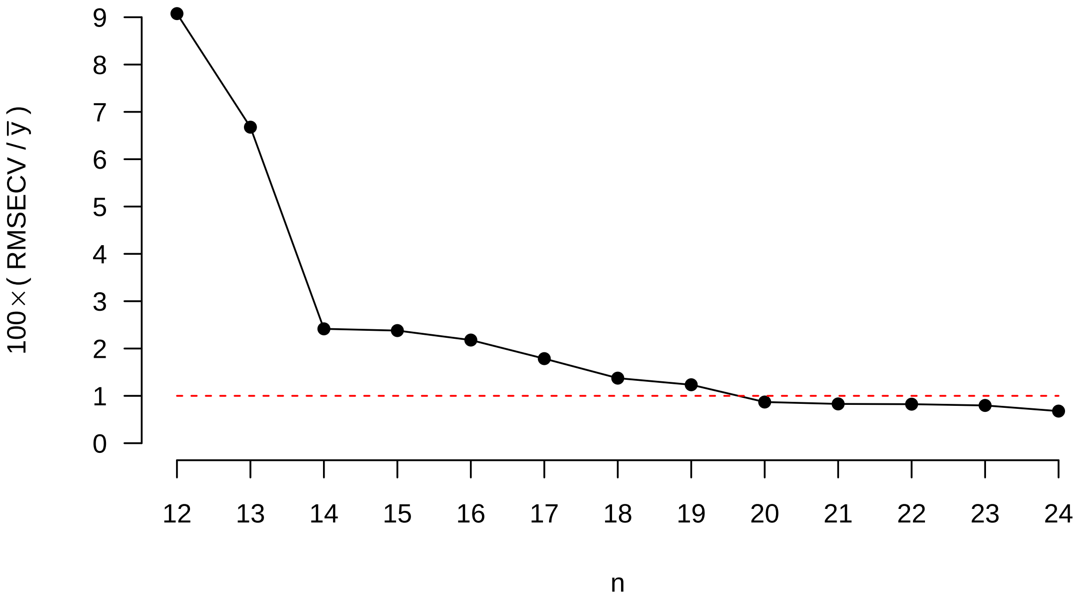

Figure 56: Percentage $RMSECV/\bar{y}$ values for the ordinary kriging models fitted using best sequential designs with $n = 12$ to 24, red dotted line shows the 1% $RMSECV/\bar{y}$.

3.7 Conclusions

In this chapter, we have proposed MaxPro designs for computer experiments involving both quantitative and qualitative factors. Unlike SLHDs, MaxPro designs can accommodate large number of qualitative factors with an economic run size. Moreover, construction of MaxPro designs is not restricted to a special design class which is the case for MCDs. More importantly, MaxPro designs are generated with an optimality criterion which enables designs can be constructed for any run size, number of quantitative factors and qualitative factors and number of levels for qualitative factors. Although MaxPro design does not require a particular design structure, we make use of OA and full factorial designs in the construction algorithm for obtaining better designs. MaxPro designs have good space-filling properties in all subspaces of the factors. We have shown their superiority in space-fillingness and Gaussian process modeling with several simulations on different cases. Performance of MaxPro designs is also illustrated on a solid end milling process optimization using its simulation. Finally, a fast sequential construction methodology is proposed for MaxPro designs. On a solid end milling process simulation, we demonstrated that the computer simulations can be analyzed with minimum resources using the sequential MaxPro designs.

3.8 Appendix

Proof.

$$\psi(\boldsymbol{D}) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j \neq i} \frac{1}{\prod_{l=1}^{p} \{(x_{il} - x_{jl})^2 + \lambda_x\}\{I(z_i \neq z_j) + \lambda_z\}}.$$

Let, for $k \neq r$, $k, r = 1, ..., t$ and $m = n/t$

$$\psi(\boldsymbol{D}_{x,k}, \boldsymbol{D}_{x,r}) = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{1}{\prod_{l=1}^{p} \{(x_{il} - x_{jl})^2 + \lambda_x\}},$$

where $\boldsymbol{D}_{x,k}$ and $\boldsymbol{D}_{x,r}$ are the sub-designs of $\boldsymbol{D}_x$ corresponding to the qualitative factor level k and r, respectively. Let C is the set with 2-combinations of the set $1, \ldots, t$.

$$\psi(\boldsymbol{D}) = \frac{1}{\binom{n}{2}} \left\{ \frac{m^2}{1+\lambda_z} \sum_{(k,r)\in C} \psi(\boldsymbol{D}_{x,k}, \boldsymbol{D}_{x,r}) + \frac{\binom{m}{2}}{\lambda_z} \sum_{i=1}^{t} \psi(\boldsymbol{D}_{x,i}) \right\}.$$

$$\psi(\boldsymbol{D}) = \frac{1}{\binom{n}{2}(1+\lambda_z)} \left\{ m^2 \sum_{(k,r)\in C} \psi(\boldsymbol{D}_{x,k}, \boldsymbol{D}_{x,r}) + \binom{m}{2} \sum_{i=1}^{t} \psi(\boldsymbol{D}_{x,i}) + \frac{\binom{m}{2}}{\lambda_z} \sum_{i=1}^{t} \psi(\boldsymbol{D}_{x,i}) \right\}.$$

$$\psi(\boldsymbol{D}) = \frac{1}{(1+\lambda_z)} \left\{ \psi(\boldsymbol{D}_x) + \frac{1}{\lambda_z} \frac{\binom{m}{2}}{\binom{n}{2}} \sum_{i=1}^{t} \psi(\boldsymbol{D}_{x,i}) \right\}.$$

◇

REFERENCES

Aceil, S. M., and Edwards, D. R. (1991),"Sensitivity Analysis of Thermal-Hydraulic Parameters and Probability Estimation of Boiling Transition in a Standard BWR/6," *Nuclear Technology*, 93, 123–130.

Audze, P., and Eglais, V. (1977), "New Approach for Planning Out Experiments," *Problems of Dynamics and Strengths*, 35, 104–107.

Ba, S. (2015), "*SLHD: Maximin-Distance (Sliced) Latin Hypercube Designs*," R package version 2.1-1, http://CRAN.R-project.org/package=SLHD

Ba, S., and Joseph, V. R. (2015), "*MaxPro: Maximum Projection Designs*," R package version 3.1-2, http://cran.r-project.org/web/packages/MaxPro

Ba, S., Brenneman, W. A., and Myers, W. R. (2015)"Optimal Sliced Latin Hypercube Designs," *Technometrics*, 57(4), 479–487

Currin, C., Mitchell, T. J., Morris, M. D., and Ylvisaker, D. (1991), "Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments," *Journal of American Statistical Association*, 86, 953–963.

Deng, X., Hung Y., and Lin C. D. (2015), "Design for Computer Experiments with Qualitative and Quantitative Factors,"*Statistica Sinica*, 25, 1567-1581.

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

Dettinger, M. D., and Wilson, J. L. (1981), "First Order Analysis of Uncertainty in Numerical Models of Groundwater Flow, Part 1. Mathematical Development," *Water Resources Research*, 17, 149–161.

DiazDelaO, F. A., Adhikari, F. A., Flores, S. S., and Friswell, E. I. (2013), "Stochastic Structural Dynamic Analysis Using Bayesian Emulators," *Computers and Structures*, 120, 24–32.

Donachie, M. J. (2000), *Titanium: A Technical Guide, 2nd Edition*, Materials Park, OH: ASM International.

Draguljic, D., Santner, T. J., and Dean, A. M. (2012), "Non-collapsing Space-filling Designs for Bounded Non-rectangular Regions," *Technometrics*, 54, 169–178.

Duan, W., Ankenman, B. E., Sanchez, S. M., and Sanchez, P. J. (2015),"Sliced Full Factorial-Based Latin Hypercube Designs as a Framework for a Batch Sequential Design Algorithm," *Technometrics*, doi:10.1080/00401706.2015.1108233.

Dwight, R. P., and Han, Z. (2009), "Efficient Uncertainty Quantification Using Gradient-Enhanced Kriging," AIAA Paper No. 2009-2276.

Fang, K. T. (1980), "The Uniform Design: Application of Number Theoretic Methods in Experimental Design," *Acta Mathematica Applicate Sinica*, 3, 363–372.

Fang, K. T., Li, R., and Sudjianto, A. (2006), *Design and Modeling for Computer Experiments*, New York: CRC Press.

Ghanem, R., and Spanos, P. (1991), *Stochastic Finite Elements: A Spectral Approach*, New York: Springer-Verlag.

Han, G., Santner, T. J., Notz, W. I., and Bartel, D. L. (2008), "Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables," *Technometrics*, 51(3), 278–288.

Helton, J. C., and Davis, F. J. (2003), "Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems, *Reliability Engineering and System Safety* 81(1), 23–69.

Helton, J. C., Johnson, J. D., Sallaberry, C. J., and Storlie, C. B. (2006), "Survey of Sampling-Based Methods for Uncertainty and Sensitivity Analysis," *Reliability Engineering and System Safety*, 91(10-11), 1175–1209.

Hickernell, F. J. (1998), "A Generalized Discrepancy and Quadrature Error Bound," *Mathematics of Computation*, 67, 299–322.

Hung, Y., Joseph, V. R., and Melkote, S. N. (2009). "Design and Analysis of Computer Experiments with Branching and Nested Factors," *Technometrics*, 51, 354–365.

Hung, Y., Joseph, V. R., and Melkote, S. N. (2015), "Analysis of Computer Experiments with Functional Response," *Technometrics*, 57 (1), 35–44.

Jin, R., Chen, and Sudjianto, A. (2005), "An Efficient Algorithm for Constructing Optimal Design of Computer Experiments," *Journal of Statistical Planning and Inference*, 134, 268–287.

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

Joseph, V. R., and Delaney, J. D. (2008), "Functionally Induced Priors for the Analysis of Experiments," *Technometrics*, 49, 1–11.

Joseph, V. R., and Hung, Y. (2008), "Orthogonal-Maximin Latin Hypercube Designs," *Statistica Sinica*, 41, 362-375.

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

Kim, H. K., Lee, Y. W., Kim, T. W., and Chang, S. H. (1986), "A procedure for Statistical Thermal Margin Analysis Using Response Surface Method and Monte Carlo Technique," *Journal of the Korean Nuclear Society*, 18, 38–47.

Lee, S. H., Kim, J. S., and Chang, S. H. (1987), "A Study on Uncertainty and Sensitivity of Operational and Modelling Parameters for Feedwater Line Break Analysis," *Journal of the Korean Nuclear Society*, 19, 10–21.

Lekivetz, R., and Jones, B. (2015), "Fast Flexible Space-Filling Designs for Nonrectangular Regions," *Quality and Reliability Engineering International*, 31 (5), 829–837.

Lewins, J., and Becker, M. (1982), *Advances in Nuclear Science and Technology: Vol. 14. Sensitivity and Uncertainty Analysis of Reactor Performance Parameters*, New York: Plenum Press.

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

McMillan, N. J., Sacks, J., Welch, W. J., and Gao, F. (1999), "Analysis of Protein Activity Data by Gaussian Stochastic Process Models," *Journal of Biopharmeceutical Statistics*, 9, 145–160.

Moon, H., Dean, A. M., and Santner, T. J. (2011), "Algorithms for Generating Maximin Latin Hypercube and Orthogonal Designs," *Journal of Statistical Theory and Practice*, 5, 81–98.

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

Owen, A. (1994), "Controlling Correlations in Latin Hypercube Samples," *Journal of American Statistical Association*, 89, 1517–1522.

Parry, G. W. (1996), "The Characterization of Uncertainty in Probabilistic Risk Assessments of Complex Systems," *Reliability Engineering and System Safety*, 54 (2-3), 11926.

Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008), "Gaussian Process Models for Computer Experiments with Qualitative and Quantitative Factors," *Technometrics*, 50, 383–396.

Qian, P. Z. G. (2012), "Sliced Lating Hypercube Designs," *Journal of the American Statistical Association*, 107(497), 393–399.

Rabitz, H., Kramer, M., and Dacol, D. (1983), "Sensitivity Analysis in Chemical Kinetics," *Annual Review of Physical Chemistry*, 34, 419–461.

Ronen, Y. (1988), *Uncertainty Analysis*, Boca Raton: CRC Press.

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

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

Shewry, M. C., and Wynn, H. P. (1987), "Maximum Entropy Sampling," *Journal of Applied Statistics*, 14, 165–170.

Tang, B., (1993), "Orthogonal Array-Based Latin Hypercubes," *Journal of American Statistical Association*, 88, 1392–1397.

Tang, D. H., and Pinder, G. F. (1977), "Simulation of Groundwater Flow and Mass Transport," *Advances in Water Resources*, 1, 25–30.

Tomovic, R., and Vukobratovic, M. (1978), *Introduction to system sensitivity theory*, New York: Academic Press.

Turányi, T. (1990), "Sensitivity Analysis of Complex Kinetic Systems, Tools and Applications," *Journal of Mathematical Chemistry*, 5, 203–248.

Wana, H. P., Mao, Z., Todd M. D., and Ren W. X. (2014)"Analytical Uncertainty Quantification for Modal Frequencies with Structural Parameter Uncertainty Using a Gaussian Process Metamodel," *Engineering Structures*, 75, 557–589.

Wu, C. F. J., and Hamada, M. (2009), *Experiments: Planning, Analysis, and Parameter Design Optimization, 2nd Edition*, New York: Wiley.

Xia, Z., Tang, J. (2013), "Characterization of Dynamic Response of Structures with Uncertainty by Using Gaussian Processes," *Journal of Vibration and Acoustics*, 135 (5), 051006-13.

Xu, J., Chen, J., and Qian, P. Z. G. (2015), "Sequentially Refined Latin Hypercube Designs: Reusing Every Point," *Journal of the American Statistical Association*, 110 (512), 1696–1706.

Ye, K. Q. (1998), "Orthogonal Column Latin Hypercube Designs and Their Application in Computer Experiments," *Journal of the American Statistical Association*, 93, 1430–1439.

Zeitoun, D. G., and Braester, C. (1991), "A Neumann Expansion Approach to Flow Through Heterogeneous Formations Stochastic Hydrology and Hydraulics," *Stochastic Hydrology and Hydraulics*, 5, 207–226.