STATISTICAL ADJUSTMENT, CALIBRATION, AND UNCERTAINTY QUANTIFICATION OF COMPLEX COMPUTER MODELS

A Thesis
Presented to
The Academic Faculty

by

Huan Yan

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 2014

Copyright © 2014 by Huan Yan

STATISTICAL ADJUSTMENT, CALIBRATION, AND UNCERTAINTY QUANTIFICATION OF COMPLEX COMPUTER MODELS

Approved by:

Roshan Joseph Vengazhiyil, Advisor
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

Jianjun Shi
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Hui Yang
Industrial & Management Systems Engineering
University of South Florida

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

Date Approved: 18 June 2014

To my beloved parents and my wife

for their support, inspiration and encouragement.

ACKNOWLEDGEMENTS

I would like thank many people who guided me, encouraged me, supported and helped me over the years through my Doctoral studies at Georgia Tech.

First and formost, with deepest appreciation, I would like to thank my advisor, Professor Roshan Joseph Vengazhiyil, for being my research mentor and his tremendous guidance, advice and immense help to complete this journey. His continual guidance and support, as well as his genuine passion for research, is inspirational and has shaped me into the researcher I am today. It is an honor and great pleasure to work for you. Thank you for imparting all your knowledge to me about the wonderful world of engineering statistics.

I also want to extend a special thanks to Professor C. F. Jeff Wu and Professor Jianjun Shi for their guidance and suggestions which pointed me in the right direction and gave me the confidence to pursue my goals. You have contributed to my success and progress in very unique and important ways.

I am thankful to Professor Shreyes N. Melkote, Professor Hui Yang, and Dr. Subramaniam Jayanti for not only being helpful to answer my questions and give advices related to my major research projects, but also for their general support and patience when things become overwhelming in difficult times.

Furthermore, I will forever be grateful to my lab mates and coworkers at and around Georgia Tech who offered their time and help on my research projects. I admire each of you and always feel blessed to work with you.

Finally, I would like to thank my wife and my parents, whose love and support I have come to rely on during this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis consists of three chapters on the statistical adjustment, calibration, and uncertainty quantification of complex computer models with applications in engineering. The first chapter systematically develops an engineering-driven statistical adjustment and calibration framework, the second chapter deals with the calibration of potassium current model in a cardiac cell, and the third chapter develops an emulator-based approach for propagating input parameter uncertainty in a solid end milling process.

Engineering model development involves several simplifying assumptions for the purpose of mathematical tractability which are often not realistic in practice. This leads to discrepancies in the model predictions. A commonly used statistical approach to overcome this problem is to build a statistical model for the discrepancies between the engineering model and observed data. In contrast, an engineering approach would be to find the causes of discrepancy and fix the engineering model using first principles. However, the engineering approach is time consuming, whereas the statistical approach is fast. The drawback of the statistical approach is that it treats the engineering model as a black box and therefore, the statistically adjusted models lack physical interpretability. In the first chapter, we propose a new framework for model calibration and statistical adjustment. It tries to open up the black box using simple main effects analysis and graphical plots and introduces statistical models inside the engineering model. This approach leads to simpler adjustment models that are physically more interpretable. The approach is illustrated using a model for predicting the cutting forces in a laser-assisted mechanical micromachining process and a model

for predicting the temperature of outlet air in a fluidized-bed process.

The second chapter studies the calibration of a computer model of potassium currents in a cardiac cell. The computer model is expensive to evaluate and contains twenty-four unknown parameters, which makes the calibration challenging for the traditional methods using kriging. Another difficulty with this problem is the presence of large cell-to-cell variation, which is modeled through random effects. We propose physics-driven strategies for the approximation of the computer model and an efficient method for the identification and estimation of parameters in this high-dimensional nonlinear mixed-effects statistical model.

Traditional sampling-based approaches to uncertainty quantification can be slow if the computer model is computationally expensive. In such cases, an easy-to-evaluate emulator can be used to replace the computer model to improve the computational efficiency. However, the traditional technique using kriging is found to perform poorly for the solid end milling process. In chapter three, we develop a new emulator, in which a base function is used to capture the general trend of the output. We propose optimal experimental design strategies for fitting the emulator. We call our proposed emulator local base emulator. Using the solid end milling example, we show that the local base emulator is an efficient and accurate technique for uncertainty quantification and has advantages over the other traditional tools.

CHAPTER I

ENGINEERING-DRIVEN STATISTICAL ADJUSTMENT AND CALIBRATION

1.1 Introduction

Models derived from physics often do not represent reality due to various simplifying assumptions made during their derivation. A commonly used approach is to collect real data from the system and estimate the unknown parameters (known as calibration parameters) in the engineering (or computer or physics-based) model. This calibration procedure brings the engineering model closer to the real observations making the model predictions more realistic. See for example Box and Hunter (1962). However, the model form itself can be wrong and therefore, merely estimating the unknown parameters from data will not eliminate the discrepancy between the predictions and the truth.

A statistical approach to overcome this problem is to build an empirical model to capture the model discrepancy. In a fundamental work, Kennedy and O'Hagan (2001) proposed to do this using Gaussian process (GP) models. Improvements to the GP modeling approach were made by Higdon et al. (2004), Bayarri et al. (2007), Qian and Wu (2008), Wang, Chen, and Tsui (2009), and Chang and Joseph (2014). Methods using linear regression models (Reese et al. 2004; Joseph and Melkote 2009) have also appeared in the literature, which again do the same job of representing the model discrepancy through a statistical model.

On the other hand, an engineering approach to deal with model discrepancy is quite different. The aim there is to understand why such a discrepancy happened and fix the wrong assumptions made in the model derivation so as to overcome the discrepancy. For example, Singh and Melkote (2009) describe a physics-based model for

predicting the cutting forces in a laser-assisted mechanical micro-machining (LAMM) process. During its development discrepancies from real data were observed. Attempts were made to improve the model, for example, by replacing an analytical model of a moving point heat source by a finite element model that solves a transient heat equation with appropriate boundary conditions and by incorporating an iterative algorithm that accounts for the deflection in the machine-tool-workpiece system.

Both approaches have pros and cons. The statistical approach is conceptually simple to implement. Once we have the real data from the system, we can fit a statistical model, be it a GP model or a linear regression model. However, such statistically-adjusted engineering model lacks predictive power in the sense that it can be used only in the experimental conditions identical or very similar to those under which the statistical adjustment model was fitted. For example, in the LAMM process, if the machine is changed, then new data need to be collected and the calibration/adjustment needs to be repeated, whereas this does not have to be done with the engineering approach. The change in the machine mainly causes a change in the stiffness which can be incorporated into the engineering model that accounts for the deflection in the machine-tool-workpiece system. Another advantage of the engineering approach is that we will have an improved understanding of the system, which will help in new developments. However, the engineering approach has a serious drawback that it is very time consuming. It takes time to hypothesize about the potential causes of discrepancies, investigate through experiments, and develop models to account for them. For example, replacing the analytical model with a finite element model in the LAMM required writing new codes which was extremely time consuming.

In this chapter we propose a new approach to model calibration and adjustment, which can be viewed as an integration of the foregoing two approaches. We call it engineering-driven statistical adjustment. The idea is to first understand about the model discrepancies using data and then to postulate statistical models that have physical interpretations. This may not sound totally new to many practitioners working in this area. What is new here is that we propose a general framework

instead of an ad hoc and problem-specific approach. Our approach can be viewed as an extension of the Kennedy-O'Hagan framework for model calibration. We start by fitting a GP model and then through main effects analysis and graphical plots we investigate the model discrepancies. This is an important step in our approach which gives statistical guidance to engineers on where to look and what to look for in the data. The next step is to postulate simple adjustment models that can account for the discrepancies. The aim here is to reduce the time needed in the development of a pure engineering model while ensuring that the statistical models have physical interpretations.

To motivate the kind of adjustment models proposed in this chapter, consider the sulphur dioxide (SO_2) mixing process discussed in Pal and Joseph (1998). In this process, the SO_2 gas is mixed with water in a tank. Using mass balancing equations of the mixing process, the concentration of the SO_2 in the output solution at time t can be obtained as

$$f(\boldsymbol{x}, t; \eta) = f_0 \exp\{-(x_1 + x_2)t/\eta\} + \frac{x_1}{x_1 + x_2}(1 - \exp\{-(x_1 + x_2)t/\eta\}),$$

where x_1 is the flow rate of SO_2 gas into the tank, x_2 is the flow rate of water into the tank, $\boldsymbol{x} = (x_1, x_2)'$, and f_0 is the initial concentration at time $t = 0$. Here the unknown parameter η represents the mass of the solution in the tank. Pal and Joseph collected data on the SO_2 concentration over time at different settings of x_1 and x_2 and found that there is a severe discrepancy between the predicted concentration by the above model and the observed data. They realized that if the model is adjusted to

$$f(\boldsymbol{x}, t; \eta) = f_0 \exp\{-(\gamma x_1 + x_2)t/\eta\} + \frac{\gamma x_1}{\gamma x_1 + x_2}(1 - \exp\{-(\gamma x_1 + x_2)t/\eta\}),$$

where γ is an unknown parameter, then the model fits the data very well. Interestingly, later it was found that the instrument used for measuring the flow rate of SO_2 gas was out-of-calibration and therefore, γ could be viewed as a coefficient introduced for correcting the scale-bias of the instrument. Pal and Joseph identified this adjustment model through trial and error, but such an approach becomes impractical when the

engineering model is complex and contains many variables. The aim of this work is to propose a systematic approach to identifying adjustment models through data analysis.

The chapter is organized as follows. In Section 1.2, we describe the main part of our engineering-driven statistical adjustment methodology (see Figure 1). We start with a simplified version of the Kennedy and O'Hagan model and try to identify the causes of discrepancy through a functional ANOVA decomposition. We then postulate statistical adjustment models to account for the discrepancy. Although this model adjustment step can be quite general, most of the time we end up in an enhanced engineering model that includes an extra set of unknown parameters. The statistical inference of such models is discussed in Section 1.3. The methodology is explained using the LAMM example in Section 1.4.1 and fluidized-bed processing example in Section 1.4.2. Some follow-up and model refinement strategies are discussed in Section 1.5 and we conclude with some remarks and future research directions in Section 1.6.

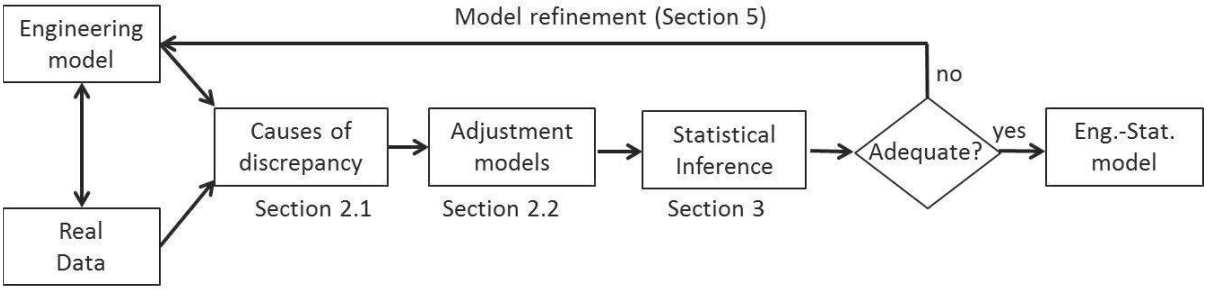

Figure 1: Engineering-driven statistical adjustment methodology.

1.2 Methodology

Let Y be the output of the system and $\boldsymbol{x} = (x_1, \cdots, x_p)'$ the input variables. Denote the engineering model by $f(\boldsymbol{x}; \boldsymbol{\eta})$, where $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_q)'$ are the calibration parameters. Then, Kennedy and O'Hagan (2001)'s model can be stated as

$$Y = \rho f(\boldsymbol{x}; \boldsymbol{\eta}) + \delta(\boldsymbol{x}) + \epsilon, \tag{1}$$

where ρ is a scale parameter, $\delta(\boldsymbol{x})$ the model bias or discrepancy function, and $\epsilon \sim^{iid} N(0, \sigma^2)$ the random error which captures the measurement noise and the effect

of unaccounted variables in the system. The scale parameter, calibration parameters, and the bias function can be estimated from the real data $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)$. Furthermore, Kennedy and O'Hagan (2001) proposed to model $\delta(\boldsymbol{x})$ using a stationary Gaussian process, which can be viewed as putting a prior on the discrepancy function:

$$\delta(\boldsymbol{x}) \sim GP(\boldsymbol{h}(\boldsymbol{x})'\boldsymbol{\mu}, \tau^2 R(\cdot)), \tag{2}$$

where $\boldsymbol{h}(\boldsymbol{x}) = (h_0(\boldsymbol{x}), \cdots, h_l(\boldsymbol{x}))'$ is a set of known functions, $\boldsymbol{\mu} = (\mu_0, \cdots, \mu_l)'$ a set of unknown regression parameters, and the covariance function is defined as $cov(\delta(\boldsymbol{x}_i), \delta(\boldsymbol{x}_j)) = \tau^2 R(\boldsymbol{x}_i - \boldsymbol{x}_j)$ with τ^2 denoting the variance.

Often the engineering models are complex and their evaluations can be expensive. In such cases, $f(\boldsymbol{x}; \boldsymbol{\eta})$ will be replaced by an easy-to-evaluate approximate model (known as metamodel, surrogate model, or emulator) fitted based on the data from a computer experiment (Sacks et al. 1989). Kennedy and O'Hagan (2001) explains how the uncertainties in the approximation using a GP model can be incorporated into the calibration. However, we will ignore these uncertainties for the moment and revisit this problem later in Section 1.3.2.

1.2.1 Understanding the Model Discrepancy

Before proceeding further, we slightly modify the Kennedy and O'Hagan's model specification by letting $\rho = 1$ and $\boldsymbol{\mu} = \mathbf{0}$ in (1) and (2). This is because, in our framework, ρ and $\boldsymbol{\mu}$ belong to what we call adjustment parameters, which if needed, can be identified at a later stage. Thus our model becomes

$$\begin{aligned} Y &= f(\boldsymbol{x}; \boldsymbol{\eta}) + \delta(\boldsymbol{x}) + \epsilon, \\ \delta(\boldsymbol{x}) &\sim GP(0, \tau^2 R(\cdot)), \end{aligned} \tag{3}$$

and $\epsilon \sim^{iid} N(0, \sigma^2)$. Note that with this simplified model, the engineering model can be viewed as the prior mean of the unknown true function.

Our first step is to estimate the model discrepancy and try to understand the causes of the discrepancy. For the discrepancy term $\delta(\boldsymbol{x})$, we use the Gaussian correlation function given by

$$R(\boldsymbol{x}_i - \boldsymbol{x}_j) = \exp\{-\sum_{k=1}^{p} \theta_k (x_{ik} - x_{jk})^2\}.$$

Thus, the unknown parameters in the model are $\boldsymbol{\eta}$, σ^2, τ^2, and $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)'$. Let $\boldsymbol{\phi} = (\sigma^2, \tau^2, \boldsymbol{\theta}')'$. Higdon et al. (2004) and Bayarri et al. (2007) describe fully Bayesian approaches for estimating them. But since we use this model only as a starting point to build our engineering-driven statistical adjustment model, a much simplified approach can be adopted. Let $p(\boldsymbol{\eta}, \boldsymbol{\phi}) = p(\boldsymbol{\eta})p(\boldsymbol{\phi})$ be a prior on the unknown parameters.

Integrating out $(\delta(\boldsymbol{x}_1), \ldots, \delta(\boldsymbol{x}_n))$ from the joint posterior, we obtain

$$p(\boldsymbol{\eta}, \boldsymbol{\phi}|\boldsymbol{y}) \propto \frac{1}{|\tau^2\boldsymbol{R} + \sigma^2\boldsymbol{I}|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{f}(\boldsymbol{\eta}))'(\tau^2\boldsymbol{R}+\sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y}-\boldsymbol{f}(\boldsymbol{\eta}))\}p(\boldsymbol{\eta}, \boldsymbol{\phi}), \quad (4)$$

where $\boldsymbol{R}$ is an $n \times n$ correlation matrix with ijth element $R(\boldsymbol{x}_i - \boldsymbol{x}_j)$, $\boldsymbol{I}$ is the $n \times n$ identity matrix, $\boldsymbol{f}(\boldsymbol{\eta}) = (f(\boldsymbol{x}_1; \boldsymbol{\eta}), \cdots, f(\boldsymbol{x}_n; \boldsymbol{\eta}))'$, and $\boldsymbol{y} = (y_1, \cdots, y_n)'$. Thus, we can obtain the estimates (posterior modes) of $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$ by maximizing $p(\boldsymbol{\eta}, \boldsymbol{\phi}|\boldsymbol{y})$. Denote the estimates by $\hat{\boldsymbol{\eta}}$ and $\hat{\boldsymbol{\phi}}$. The posterior mean of the discrepancy function is given by

$$\hat{\delta}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})' \left(\boldsymbol{R} + \frac{\sigma^2}{\tau^2}\boldsymbol{I}\right)^{-1} (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\eta})), \quad (5)$$

where $\boldsymbol{r}(\boldsymbol{x}) = (R(\boldsymbol{x} - \boldsymbol{x}_1), \cdots, R(\boldsymbol{x} - \boldsymbol{x}_n))'$. It can be computed by plugging-in the estimates of $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$.

We now perform a functional ANOVA decomposition of $\hat{\delta}(\boldsymbol{x})$ as described in Sacks et al. (1989) and Welch et al. (1992). More details about functional ANOVA and sensitivity analysis can be found in Santner, Williams, and Notz (2003, ch. 7) and Fang, Li, and Sudjianto (2006, ch. 6). Assume that all the variables are scaled in $[0, 1]^p$. Then, the mean effect can be computed as

$$\hat{\delta}_0 = \int_{[0,1]^p} \hat{\delta}(\boldsymbol{x})\, \boldsymbol{dx},$$

and the main effects as

$$\hat{\delta}_i(x_i) = \int_{[0,1]^{p-1}} \hat{\delta}(\boldsymbol{x})\, \boldsymbol{dx}_{-i} - \hat{\delta}_0,$$

for $i = 1, \cdots, p$. Since we use a Gaussian correlation function, explicit expressions of these quantities can be obtained as in Oakley and O'Hagan (2004) and Chen, Jin,

and Sudjianto (2005). Let $\boldsymbol{c} = (\boldsymbol{R} + \sigma^2/\tau^2 \boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{\eta}))$. Then,

$$\begin{aligned}\hat{\delta}_0 &= \sum_{j=1}^{n} c_j \prod_{k=1}^{p} \int_0^1 e^{-\theta_k (x_k - x_{kj})^2} dx_k \\ &= \sum_{j=1}^{n} c_j \prod_{k=1}^{p} \sqrt{\frac{\pi}{\theta_k}} [\Phi(\sqrt{2\theta_k}(1 - x_{kj})) - \Phi(-\sqrt{2\theta_k} x_{kj})], \end{aligned} \tag{6}$$

where $\Phi(\cdot)$ is the standard normal distribution function. Similarly,

$$\begin{aligned}\hat{\delta}_i(x_i) &= \sum_{j=1}^{n} c_j e^{-\theta_i (x_i - x_{ij})^2} \prod_{k \neq i}^{p} \int_0^1 e^{-\theta_k (x_k - x_{kj})^2} dx_k - \hat{\delta}_0 \\ &= \sum_{j=1}^{n} c_j e^{-\theta_i (x_i - x_{ij})^2} \prod_{k \neq i}^{p} \sqrt{\frac{\pi}{\theta_k}} [\Phi(\sqrt{2\theta_k}(1 - x_{kj})) - \Phi(-\sqrt{2\theta_k} x_{kj})] - \hat{\delta}_0. \end{aligned} \tag{7}$$

It is also possible to compute the two-factor and higher order interactions, but they are much harder to interpret. Therefore, we propose to gather useful information about the model discrepancy using only the main effects. For doing this main effects analysis, we rely on the effect hierarchy principle that most physical systems are governed by the lower order effects and the effect heredity principle that in case there are significant higher order effects, it will be seen in their lower order effects as well (Hamada and Wu 1992). However, to make sure that we are not missing anything important, we also compute the global sensitivity indices (Sobol 2001) of the higher order effects (see Section 1.4). In fact, a computationally simpler alternative is to compute the total sensitivity indices of each variable (see Fang et al, 2006, pp. 196) and compare them with the corresponding main effect indices.

We can now plot $\hat{\delta}_i(x_i)$ against each x_i and identify the most important variables affecting the model discrepancy. From this we can try to investigate the causes for the observed discrepancy. This step requires engineering knowledge on that particular system. We may identify more than one potential cause, but collectively they will provide insights for postulating adjustment models as described in the next subsection.

1.2.2 Postulating Adjustment Models

After identifying the potential causes for discrepancy, there are two ways to proceed. We could directly work on the engineering model by correcting the model

assumptions and by incorporating additional variables that would potentially explain the causes of discrepancy. As explained in the introduction, this engineering approach can be time consuming. Instead, we propose to postulate simple statistical models to address the potential causes of discrepancy. These statistical models are again identified through some graphical plots as described below.

First perform the main effect analysis on the calibrated engineering model $f(\boldsymbol{x}; \hat{\boldsymbol{\eta}})$. One efficient approach to do this is to use Quasi-Monte Carlo (QMC) sampling (see, e.g., Saltelli 2002). Let $\{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}\}$ be a QMC sample from $[0,1]^p$. Then, the overall mean effect is given by $f_o = \sum_{j=1}^{m} f(\boldsymbol{x}^{(j)}; \hat{\boldsymbol{\eta}})/m$ and the main effects

$$f_i(x_i) = \frac{1}{m} \sum_{j=1}^{m} f((\boldsymbol{x}_{(i)}^{(j)}, x_i); \hat{\boldsymbol{\eta}}) - f_0,$$

for $i = 1, \cdots, p$, where $(\boldsymbol{x}_{(i)}^{(j)}, x_i)$ denotes the vector $\boldsymbol{x}^{(j)}$ with its ith element replaced by x_i.

The bias-corrected engineering model based on the data is given by

$$\hat{y}(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\eta}) + \hat{\delta}(\boldsymbol{x}).$$

Because we have already performed functional ANOVA decomposition on the two terms separately, we can easily obtain the mean effect of $\hat{y}(\boldsymbol{x})$ as $\hat{y}_0 = f_0 + \hat{\delta}_0$ and the main effects as

$$\hat{y}_i(x_i) = f_i(x_i) + \hat{\delta}_i(x_i). \tag{8}$$

Note that $f_i(x_i)$ shows the effect of x_i on the engineering model, whereas $\hat{y}_i(x_i)$ shows the effect of x_i on the engineering model after correcting for the bias based on the observed data. Thus, we can plot $f_i(x_i)$ and $\hat{y}_i(x_i)$ against x_i side-by-side and try to understand the changes in the main effects due to the discrepancy. To simplify this procedure, we only need to look at the plots of the variables identified in the previous section having significant effects on the model discrepancy. These plots will help engineers and statisticians postulate meaningful statistical models to account for the observed discrepancy. This step requires engineering knowledge of the particular system under investigation, but we provide some general guidelines below.

Let $\boldsymbol{\gamma}$ be the new unknown parameters in the statistical adjustment model. We will call them *adjustment parameters*. Thus, denote the statistically-adjusted engineering model by $g(\boldsymbol{x}; \boldsymbol{\eta}, \boldsymbol{\gamma})$. Based on our experience, an adjustment model that is especially useful is a scale adjustment model given by

$$g(\boldsymbol{x}; \boldsymbol{\eta}, \boldsymbol{\gamma}) = f(\gamma_1 x_1, \cdots, \gamma_p x_p; \boldsymbol{\eta}). \tag{9}$$

They can be identified from the side-by-side plots of $f_i(x_i)$ and $\hat{y}_i(x_i)$ against x_i. It is not necessary to introduce p new parameters; we only need to include parameters for variables exhibiting discrepancy identified from the graphical plots. For example, if $\hat{y}_1(x_1)$ looks tilted compared to $f_1(x_1)$, then we can include γ_1 in the model (see the top left panel of Figure 5 for an example). But if the amount of tilting changes with respect to x_1, then including γ_1 alone cannot take care of the discrepancy. Thus, we may need to consider a more elaborate scale-power model given by

$$g(\boldsymbol{x}; \boldsymbol{\eta}, \boldsymbol{\gamma}) = f(\gamma_{10} x_1^{\gamma_{11}}, \cdots, \gamma_{p0} x_p^{\gamma_{p1}}; \boldsymbol{\eta}). \tag{10}$$

The foregoing scale and scale-power models should be used only with nonnegative variables. It is important to note that the scale-power transformations are applied on the original scale of the variables. Thus, if these variables are standardized in $[0, 1]^p$ for the purpose of model fitting, then they should be transformed back to their original scales before fitting (9) and (10). When unrestricted variables are included in the engineering model (such as angle and temperature), we can consider a location-scale model

$$g(\boldsymbol{x}; \boldsymbol{\eta}, \boldsymbol{\gamma}) = f(\gamma_{01} + \gamma_{11} x_1, \cdots, \gamma_{p0} + \gamma_{p1} x_p; \boldsymbol{\eta}). \tag{11}$$

Of course, combinations of location-scale and scale-power models can be considered depending on the type of variables and their boundary conditions.

Although (9), (10), and (11) are simple statistical adjustments to the engineering models, surprisingly they are found to be powerful in dealing with a variety of engineering systems. Moreover, many times, some interpretations to the adjustment parameters $\boldsymbol{\gamma}$ can also be given (like the SO_2 example in the introduction). Another advantage of these adjustment models is that they tend to keep the shape of the

original engineering model such as monotonicity, which have some physical meaning. Such properties are often lost when a GP model adjustment is used. This is a point noted by Joseph and Melkote (2009), where they tried to find minimal adjustment models by applying variable selection on linear regression models. The adjustment models introduced here are more suitable for making the minimal adjustment. Another advantage of the proposed adjustment models is that they are more capable in extrapolation. The GP model adjustment goes back to the mean when extrapolating, whereas the trends identified in the proposed adjustment are retained into regions outside the data range as well (see the surface roughness example in Joseph and Melkote 2009).

We want to emphasize that the foregoing adjustment models should be used only as guidelines and not as a norm for statistical adjustment. A completely different form of the adjustment may be suitable for a particular problem. For example, for overcoming the discrepancy in the force profiles observed over the immersion angle (x) in a micro-milling process, Kumar et al. (2013) considered an adjustment model of the form $\gamma_0 + \gamma_1 \sin x + \gamma_2 \cos x$. This is because such an adjustment model can be viewed as correcting for the spindle run out in the micro-milling machine and thus has a physical interpretation.

1.3 Statistical Inference

Let $\boldsymbol{\alpha} = (\boldsymbol{\eta}', \boldsymbol{\gamma}')'$ be the complete set of unknown parameters in our new engineering-statistical model. We can estimate them by fitting:

$$y_i = g(\boldsymbol{x}_i; \boldsymbol{\alpha}) + e_i + \epsilon_i, \tag{12}$$

for $i = 1, \cdots, n$, where $\epsilon_i \sim^{iid} N(0, \sigma^2)$ is the measurement error as before and $e_i \sim^{iid} N(0, \sigma_e^2)$ is the residual model bias (i.e., the bias not captured by $g(\cdot;\cdot)$). This is just a nonlinear regression model, however problems can arise in fitting the model, particularly when the number of unknown parameters is large compared to the available data. There may be insufficient engineering knowledge to choose the potential causes based on the graphical analysis and one may have to postulate an

unnecessarily large adjustment model. Thus, many adjustment parameters may do the same job of explaining the discrepancy leading to the multicollinearity problem in regression analysis. This can be avoided using shrinkage estimation as in ridge regression. In a Bayesian framework, shrinkage can be achieved by putting a prior on the adjustment parameters. As demonstrated by Kennedy and O'Hagan (2001), the Bayesian framework has the added advantage that it can easily incorporate various sources of uncertainty such as those present in approximating a computationally expensive engineering model. Therefore, we follow a Bayesian approach and develop the necessary estimation and uncertainty quantification techniques for the proposed adjustment models in the following subsections.

1.3.1 Prior Specification

Putting a prior on the adjustment parameters is not trivial because of the nonlinearity of the engineering/adjustment model. In a linear regression model this was easy to do as the variables can be standardized to obtain γ_i's in the same scale. This is no longer true with a nonlinear regression model. To obtain a meaningful prior for the adjustment parameters, we extend the functionally induced prior construction for linear regression models introduced in Joseph (2006). This approach fits in very well with our problem because we already have a GP prior (3) placed on the underlying true function. To do this, let $\boldsymbol{\gamma}_0$ be such that $g(\boldsymbol{x};\boldsymbol{\eta},\boldsymbol{\gamma}_0) = f(\boldsymbol{x};\boldsymbol{\eta})$. For example, in a scale adjustment model $\boldsymbol{\gamma}_0 = (1,\cdots,1)'$. Using Taylor series expansion

$$g(\boldsymbol{x};\boldsymbol{\alpha}) \approx f(\boldsymbol{x};\boldsymbol{\eta}) + (\boldsymbol{\gamma}-\boldsymbol{\gamma}_0)' \left[\frac{\partial}{\partial \boldsymbol{\gamma}} g(\boldsymbol{x};\boldsymbol{\eta},\boldsymbol{\gamma})\right]_{\boldsymbol{\gamma}=\boldsymbol{\gamma}_0}.$$

Let $\boldsymbol{g}(\boldsymbol{\alpha}) = (g(\boldsymbol{x}_1;\boldsymbol{\alpha}),\cdots,g(\boldsymbol{x}_n;\boldsymbol{\alpha}))'$ and $\boldsymbol{G}_0(\boldsymbol{\eta})$ be a matrix whose ijth element is $\frac{\partial}{\partial \gamma_j} g(\boldsymbol{x}_i;\boldsymbol{\eta},\boldsymbol{\gamma})$ evaluated at $\boldsymbol{\gamma}=\boldsymbol{\gamma}_0$. Then,

$$\boldsymbol{g}(\boldsymbol{\alpha}) \approx \boldsymbol{f}(\boldsymbol{\eta}) + \boldsymbol{G}_0(\boldsymbol{\eta})(\boldsymbol{\gamma}-\boldsymbol{\gamma}_0),$$

for $\boldsymbol{\gamma}$ close to $\boldsymbol{\gamma}_0$. Previously we have placed a GP prior on the discrepancy function: $\delta(\boldsymbol{x}) \sim GP(0,\tau^2 R(\cdot))$. Thus, we have $\boldsymbol{\delta} = (\delta(\boldsymbol{x}_1),\cdots,\delta(\boldsymbol{x}_n))' \sim N(\boldsymbol{0},\tau^2\boldsymbol{R})$. Now we place a prior on $\boldsymbol{\gamma}$ such that $\boldsymbol{g}(\boldsymbol{\alpha}) - \boldsymbol{f}(\boldsymbol{\eta})$ can approximate the prior on $\boldsymbol{\delta}$. Solving

for $\boldsymbol{\gamma}$ using generalized least squares, we obtain

$$\boldsymbol{\gamma} = \boldsymbol{\gamma}_0 + \{\boldsymbol{G}_0(\boldsymbol{\eta})'\boldsymbol{R}^{-1}\boldsymbol{G}_0(\boldsymbol{\eta})\}^{-1}\boldsymbol{G}_0(\boldsymbol{\eta})'\boldsymbol{R}^{-1}(\boldsymbol{g}(\boldsymbol{\alpha}) - \boldsymbol{f}(\boldsymbol{\eta})).$$

By neglecting $e_1, \ldots, e_n$, we have $\boldsymbol{g}(\boldsymbol{\alpha}) - \boldsymbol{f}(\boldsymbol{\eta}) \sim N(\boldsymbol{0}, \tau^2\boldsymbol{R})$ approximately. Thus, we obtain

$$\boldsymbol{\gamma}|\boldsymbol{\eta} \sim N(\boldsymbol{\gamma}_0, \tau^2\{\boldsymbol{G}_0(\boldsymbol{\eta})'\boldsymbol{R}^{-1}\boldsymbol{G}_0(\boldsymbol{\eta})\}^{-1}). \tag{13}$$

1.3.2 Posterior Analysis

We simplify the posterior analysis by fixing certain parameters, specifically $\boldsymbol{\phi} = (\sigma^2, \tau^2, \boldsymbol{\theta}')'$ is fixed at its posterior mode $\hat{\boldsymbol{\phi}}$ obtained earlier. Thus, our objective is to find the posterior distribution of $\boldsymbol{\alpha} = (\boldsymbol{\eta}', \boldsymbol{\gamma}')'$ and σ_e^2 given $\boldsymbol{\phi} = \hat{\boldsymbol{\phi}}$. It can be obtained as

$$\begin{aligned} p(\boldsymbol{\alpha}, \sigma_e^2|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\alpha}, \sigma_e^2)p(\boldsymbol{\alpha})p(\sigma_e^2) \\ &\propto \frac{1}{(\hat{\sigma}^2 + \sigma_e^2)^{n/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\frac{[y_i - g(\boldsymbol{x}_i;\boldsymbol{\alpha})]^2}{(\hat{\sigma}^2 + \sigma_e^2)}\right\} p(\boldsymbol{\gamma}|\boldsymbol{\eta})p(\boldsymbol{\eta})p(\sigma_e^2). \end{aligned} \tag{14}$$

Clearly, we cannot obtain an explicit expression for the posterior distribution. We can use MCMC methods to sample from the posterior and compute the desired posterior quantities. Let $(\boldsymbol{\alpha}_j, \sigma_{ej}^2)$, $j = 1, \ldots, m$ be the posterior sample. Then, the mean prediction at any $\boldsymbol{x}$ can be obtained as

$$\bar{g}(\boldsymbol{x}) = \frac{1}{m}\sum_{j=1}^{m} g(\boldsymbol{x};\boldsymbol{\alpha}_j).$$

A credible interval for the mean prediction can be obtained by computing appropriate quantiles of $g(\boldsymbol{x};\boldsymbol{\alpha}_j)$ at each $\boldsymbol{x}$. Prediction intervals can be similarly computed by sampling from $N(g(\boldsymbol{x};\boldsymbol{\alpha}_j), \sigma^2 + \sigma_{ej}^2)$ for $j = 1, \ldots, m$.

The estimation and uncertainty quantification become much more complex when a metamodel is used for approximating a computationally expensive engineering model. This is because we also need to account for the uncertainties in the metamodel. A fully Bayesian treatment of this problem is discussed in Higdon et al. (2004), which can be computationally demanding. Following Chang and Joseph (2014), here we use a much simpler approach. As in the previous works (Kennedy and O'Hagan

2001, Bayarri et al. 2007), assume that the metamodel for the engineering model is constructed using only the data from a computer experiment. Let $\hat{f}(\boldsymbol{x};\boldsymbol{\eta})$ be the metamodel and $s^2(\boldsymbol{x};\boldsymbol{\eta})$ be a variance estimate for the prediction at $(\boldsymbol{x},\boldsymbol{\eta})$ using the metamodel. Given the computer experiment data, assume that

$$\begin{aligned} f(\boldsymbol{x};\boldsymbol{\eta}) &= \hat{f}(\boldsymbol{x};\boldsymbol{\eta}) + s(\boldsymbol{x},\boldsymbol{\eta})r(\boldsymbol{x};\boldsymbol{\eta}), \\ r(\boldsymbol{x};\boldsymbol{\eta}) &\sim^{iid} N(0,1). \end{aligned} \tag{15}$$

Of course it is more accurate to assume a smooth GP for $r(\boldsymbol{x};\boldsymbol{\eta})$, but recent results show that the correlations in a residual process are small (Haaland and Qian 2011, Ba and Joseph 2012). Ignoring the correlations in $r(\boldsymbol{x};\boldsymbol{\eta})$ leads to tremendous computational simplifications as we show below.

First note that $f(\boldsymbol{\eta}) \sim N(\hat{f}(\boldsymbol{\eta}), \boldsymbol{S}(\boldsymbol{\eta}))$, where $\hat{f}(\boldsymbol{\eta}) = (\hat{f}(\boldsymbol{x}_1;\boldsymbol{\eta}),\ldots,\hat{f}(\boldsymbol{x}_n;\boldsymbol{\eta}))'$ and $\boldsymbol{S}(\boldsymbol{\eta}) = diag\{s^2(\boldsymbol{x}_1;\boldsymbol{\eta}),\ldots,s^2(\boldsymbol{x}_n;\boldsymbol{\eta})\}$. Now by integrating out $\boldsymbol{f}(\boldsymbol{\eta})$ from (4), we obtain

$$p(\boldsymbol{\eta},\boldsymbol{\phi}|\boldsymbol{y}) \propto \frac{1}{|\tau^2\boldsymbol{R}+\boldsymbol{S}(\boldsymbol{\eta})+\sigma^2\boldsymbol{I}|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{y}-\hat{\boldsymbol{f}}(\boldsymbol{\eta}))'(\tau^2\boldsymbol{R}+\boldsymbol{S}(\boldsymbol{\eta})+\sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y}-\hat{\boldsymbol{f}}(\boldsymbol{\eta}))\} p(\boldsymbol{\eta},\boldsymbol{\phi}). \tag{16}$$

Maximizing $p(\boldsymbol{\eta},\boldsymbol{\phi}|\boldsymbol{y})$ with respect to $\boldsymbol{\eta}$ and $\boldsymbol{\phi}$, we can obtain their estimates $\hat{\boldsymbol{\eta}}$ and $\hat{\boldsymbol{\phi}}$. Now proceeding as before, we can identify the adjustment model $g(\boldsymbol{x};\boldsymbol{\alpha})$. For simplicity, consider a scale adjustment model, which can be written as $g(\boldsymbol{x};\boldsymbol{\alpha}) = f(\boldsymbol{x}\odot\boldsymbol{\gamma};\boldsymbol{\eta})$, where $\odot$ represents element-wise multiplication. Let $\hat{g}(\boldsymbol{x};\boldsymbol{\alpha}) = \hat{f}(\boldsymbol{x}\odot\boldsymbol{\gamma};\boldsymbol{\eta})$ and $\hat{\boldsymbol{G}}_0(\boldsymbol{\eta})$ be the gradient matrix obtained using $\hat{g}(\boldsymbol{x};\boldsymbol{\alpha})$. Fixing $\boldsymbol{G}_0(\boldsymbol{\eta})$ at $\widehat{\boldsymbol{G}}_0(\boldsymbol{\eta})$, we can integrate out $\boldsymbol{f}(\boldsymbol{\eta})$ from the unnormalized posterior in (14) to obtain (with $\boldsymbol{\phi} = \hat{\boldsymbol{\phi}}$)

$$\begin{aligned} p(\boldsymbol{\alpha},\sigma_e^2|\boldsymbol{y}) \propto\ & \prod_{i=1}^{n}[\hat{\sigma}^2+\sigma_e^2+s^2(\boldsymbol{x}_i\odot\boldsymbol{\gamma};\boldsymbol{\eta})]^{-1/2} \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\frac{[y_i-\hat{g}(\boldsymbol{x}_i;\boldsymbol{\alpha})]^2}{[\hat{\sigma}^2+\sigma_e^2+s^2(\boldsymbol{x}_i\odot\boldsymbol{\gamma};\boldsymbol{\eta})]}\right\} \\ & \times p(\boldsymbol{\gamma}|\boldsymbol{\eta})p(\boldsymbol{\eta})p(\sigma_e^2). \end{aligned} \tag{17}$$

Now we can proceed exactly as in the previous subsection to generate an MCMC sample and compute the necessary posterior quantities.

Note that by virtue of the assumption in (15), the mean prediction at $\boldsymbol{x}$ simplifies

to

$$\begin{aligned}\bar{g}(\boldsymbol{x}) &= E\{g(\boldsymbol{x};\boldsymbol{\alpha})|\boldsymbol{y}\} = E[E\{g(\boldsymbol{x};\boldsymbol{\alpha})|\boldsymbol{\alpha},\boldsymbol{y}\}|\boldsymbol{y}] \\ &= E[\hat{g}(\boldsymbol{x};\boldsymbol{\alpha})|\boldsymbol{y}].\end{aligned}$$

It can be estimated from the MCMC sample as $\bar{g}(\boldsymbol{x}) = \sum_{j=1}^{m} \hat{g}(\boldsymbol{x};\boldsymbol{\alpha}_j)/m$. The prediction intervals can be computed by sampling from $N(\hat{g}(\boldsymbol{x};\boldsymbol{\alpha}_j), \hat{\sigma}^2 + \sigma_{ej}^2 + s^2(\boldsymbol{x} \odot \boldsymbol{\gamma}_j;\boldsymbol{\eta}_j))$ for $j = 1,\ldots,m$. We can see that the variance is inflated because of the uncertainties in the metamodel and therefore, the prediction intervals will become wider.

1.4 Examples

1.4.1 LAMM Example

Consider the laser-assisted mechanical micromachining (LAMM) process discussed in the introduction. The engineering models for this process were developed in Singh and Melkote (2009) and calibrated using real data in Singh, Joseph, and Melkote (2011). The engineering model is a highly complex computer model consisting of a geometrical model to compute strain and strain rates, a finite element model to compute the temperature distribution, a material model for computing the stresses, a force model for computing the forces, and an iterative algorithm to account for the machine-tool-workpiece deflection.

Four variables: nominal depth of cut (x_1), cutting speed (x_2), laser power (x_3), and laser location (x_4) are selected for process optimization. Because it takes about 14 hours for a single evaluation of the computer model, an easy-to-evaluate approximation of the model is needed. For this purpose, a full factorial $4 \times 2 \times 3 \times 2$ computer experiment was performed. Several outputs were obtained from the computer model, but for simplicity, here we analyze only the cutting forces (y). Singh et al. (2011) proposed the following nonlinear regression model to approximate the computer model:

$$f(\boldsymbol{x}) = \beta_0 x_1^{\beta_1} \exp\left\{\beta_2 x_2 - \beta_3 x_3 e^{-\beta_4 x_4}\right\}. \tag{18}$$

The nonlinear regression fitting gave $\hat{\beta}_0 = 1.3586, \hat{\beta}_1 = 0.8887, \hat{\beta}_2 = 0.0014, \hat{\beta}_3 =$

0.0268, and $\hat{\beta}_4 = 0.0034$ with an $R^2 = 0.9972$ and residual standard error $\hat{\sigma}_f = 0.2749$. Because of the excellent fit and the model's several physical interpretations, the above nonlinear regression was preferred over the commonly used kriging metamodel.

A physical experiment was also carried out on the LAMM process using the same 48-run experimental design but with three replicates per run. Figure 2 plots the (exact) computer model predictions and the average cutting force measured in the physical process averaged over the three replicates. We can see that most of the values lie above the 45^o line indicating that the engineering model under-predicts the cutting forces. Thus, there is a severe model discrepancy. There can be several reasons for this discrepancy. It can be due to assuming the specific heat coefficient used in the thermal model to be temperature independent or due to neglecting the plasticity effects in the geometrical model or due to using wrong coefficients in the Johnson-Cook material model, and so on. As discussed in the introduction, a pure engineering approach of checking all these possibilities, identifying the correct cause, implementing and validating the model, is a time consuming process. On the other hand, building a statistical model to account for discrepancy as in Kennedy and O'Hagan (2001) is easy, but it lacks physical interpretations. Below we explain how the model can be corrected using the proposed engineering-driven statistical adjustment approach.

First the main effects from the functional ANOVA decomposition of the meta-model are computed using QMC sampling and are plotted in Figure 3(a). We can see that the nominal depth of cut (x_1) has the largest effect on the cutting force followed by the laser power (x_3). Now we can fit a GP model to the model discrepancy as described in Section 1.2.1. There were three replicates in each run of the experiment. These were used to compute the error variance as 1.42, which has degrees of freedom $48 \times (3-1) = 96$. We will analyze the average of the three observations and therefore, the error variance should be 1.42/3. We use this value to specify an inverse-Gamma prior for σ^2 as $\sigma^2 \sim IG(96/2, (96/2+1) \times 1.42/3)$, where the parameters are chosen such that the mode is at 1.42/3. For rest of the parameters, we use a noninformative prior $p(\tau^2, \boldsymbol{\theta}) \propto 1$. The prediction variance (see, Bates and Watts 1988) from the

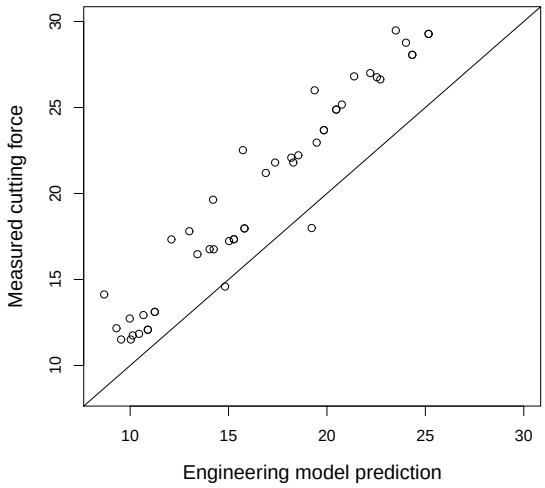

Figure 2: Plot of the cutting force predictions from the engineering model and the measured values.

nonlinear regression is given by

$$s^2(\boldsymbol{x}) = \sigma_f^2 + \sigma_f^2 \dot{\boldsymbol{f}}(\boldsymbol{x})'(\dot{\boldsymbol{F}}'\dot{\boldsymbol{F}})^{-1}\dot{\boldsymbol{f}}(\boldsymbol{x}),$$

where $\dot{\boldsymbol{f}}(\boldsymbol{x})$ is the gradient vector of $\hat{f}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ and $\dot{\boldsymbol{F}} = (\dot{\boldsymbol{f}}(\boldsymbol{x}_1), \cdots, \dot{\boldsymbol{f}}(\boldsymbol{x}_n))'$. Because $\hat{f}(\boldsymbol{x})$ has a simple form, we can easily obtain

$$\dot{\boldsymbol{f}}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) \left(\frac{\hat{\beta}_1}{x_1}, \hat{\beta}_2, -\hat{\beta}_3 e^{-\hat{\beta}_4 x_4}, \hat{\beta}_3 \hat{\beta}_4 x_3 e^{-\hat{\beta}_4 x_4} \right)'.$$

Now by maximizing $p(\boldsymbol{\phi}|\boldsymbol{y})$ in (16), we obtain the correlation parameters $\hat{\boldsymbol{\theta}} = (2.10, 0.15, 2.04, 0.21)'$ and the variance parameters $\hat{\sigma}^2 = .49$ and $\hat{\tau}^2 = 7.58$. By plugging-in these values, we can compute the discrepancy function given in (5).

We now perform the functional ANOVA decomposition of $\hat{\delta}(\boldsymbol{x})$. The main effects can be computed explicitly using (7) and are plotted in Figure 3(b). We can see that the main contributors to the discrepancy are the nominal depth of cut (x_1) and the laser power (x_3). Although we will focus on the main effects analysis, as a cautionary step, we will also compute the sensitivity indices of the two-factor interactions. The global sensitivity indices of the main effects and two-factor interactions are plotted (after a log-transform) in Figure 4. We can see that the large sensitivity indices are

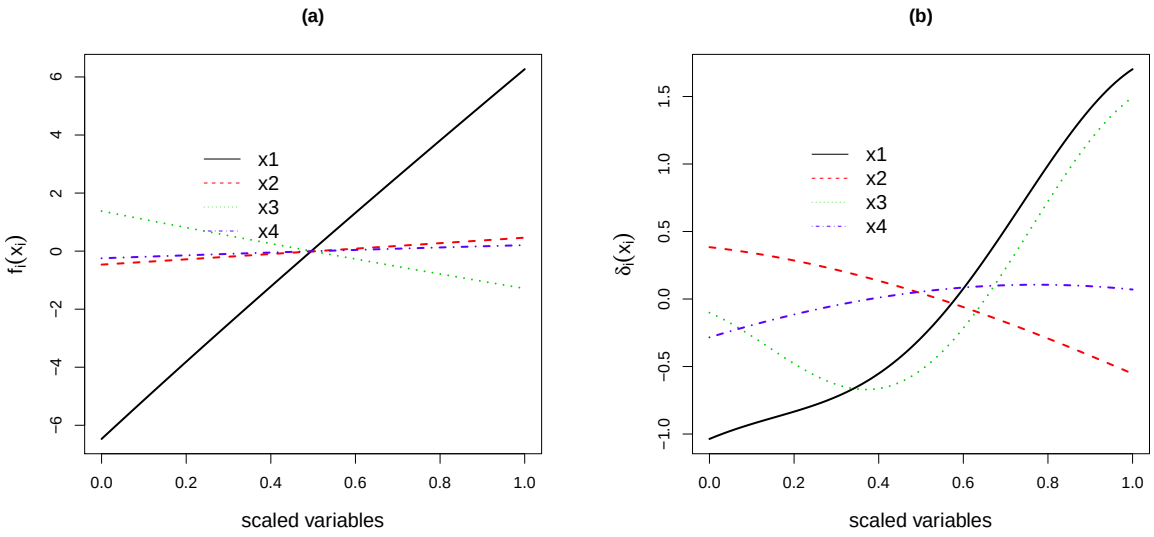

Figure 3: Main effects from the functional ANOVA decomposition of (a) $f(\boldsymbol{x})$ and (b) $\delta(\boldsymbol{x})$.

due to the main effects of x_1 and x_3. The sensitivity indices of two-factor interactions are very small and thus, a main effects analysis is reasonable in this problem. If we were to detect some large two-factor interactions, then some follow-up strategies would be needed, which will be discussed in the next section.

The next step would be to investigate why these model discrepancies are observed, especially focusing on the effects of the nominal depth of cut and laser power. In some cases, the engineers working in the process might be able to quickly point out a reason and then we can proceed to build an engineering or a statistical model to address the cause. In some other cases, the process is so complex that the engineers may not have enough knowledge to attribute the discrepancies to a particular cause. In such cases, we can perform the following analysis to develop a statistical adjustment model.

We can use (8) to quickly compute the main effects of the engineering model plus the discrepancy function. These are plotted in Figure 5 along with the main effects of the engineering model. We can see that when we use the data, the sensitivity of x_1 has increased, whereas the sensitivity of x_3 has decreased. The other two variables do not have much effect, which is expected based on the main effect plots in Figure

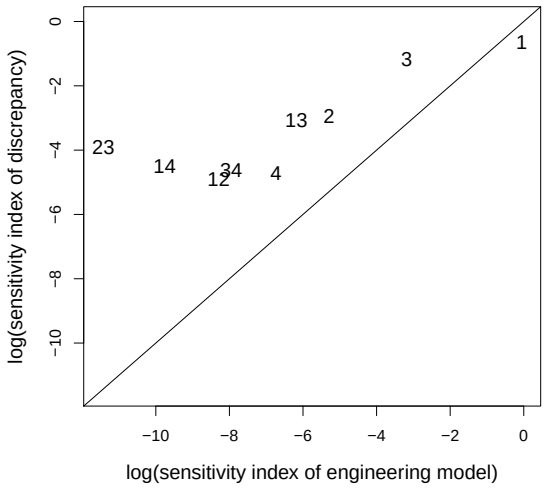

Figure 4: Plot of the log-sensitivity indices of engineering model and discrepancy.

3(b). The effect of data is to slightly tilt the main effect of x_1 and therefore, a scale adjustment model on x_1 might work. On the other hand, there is a nonlinear tilting on the main effect of x_3 and therefore, a simple scale adjustment would not be sufficient to completely capture this discrepancy. However, such a nonlinear tilting does not make any physical sense because the cutting forces are expected to decrease monotonically as the laser power increases. Thus, the following adjustment model looks physically meaningful

$$\hat{g}(\boldsymbol{x};\boldsymbol{\gamma}) = \hat{f}(\gamma_1 x_1, x_2, \gamma_3 x_3, x_4). \tag{19}$$

We also note that $\hat{y}_0 = 16.6 + 3.3 = 19.9$ is much larger than $\hat{f}_0 = 16.6$ and therefore, an adjustment model of the form $g(\boldsymbol{x};\boldsymbol{\gamma}) = \gamma_0 + \hat{f}(\boldsymbol{x})$ can describe a major part of the observed discrepancy. This is also quite clear from the scatter plot in Figure 2. However, such an intercept adjustment to the engineering model is not physically meaningful because the cutting forces are expected to decrease to 0 as $x_1 \to 0$ as well as when $x_3 \to \infty$. One might also be tempted to correct the bias using an overall scale adjustment $g(\boldsymbol{x};\boldsymbol{\gamma}) = \gamma_0 \hat{f}(\boldsymbol{x})$. Although such an adjustment is physically meaningful, it compounds the effects of many variables into one. As is quite evident from the metamodel (23), the same scale adjustment can be essentially

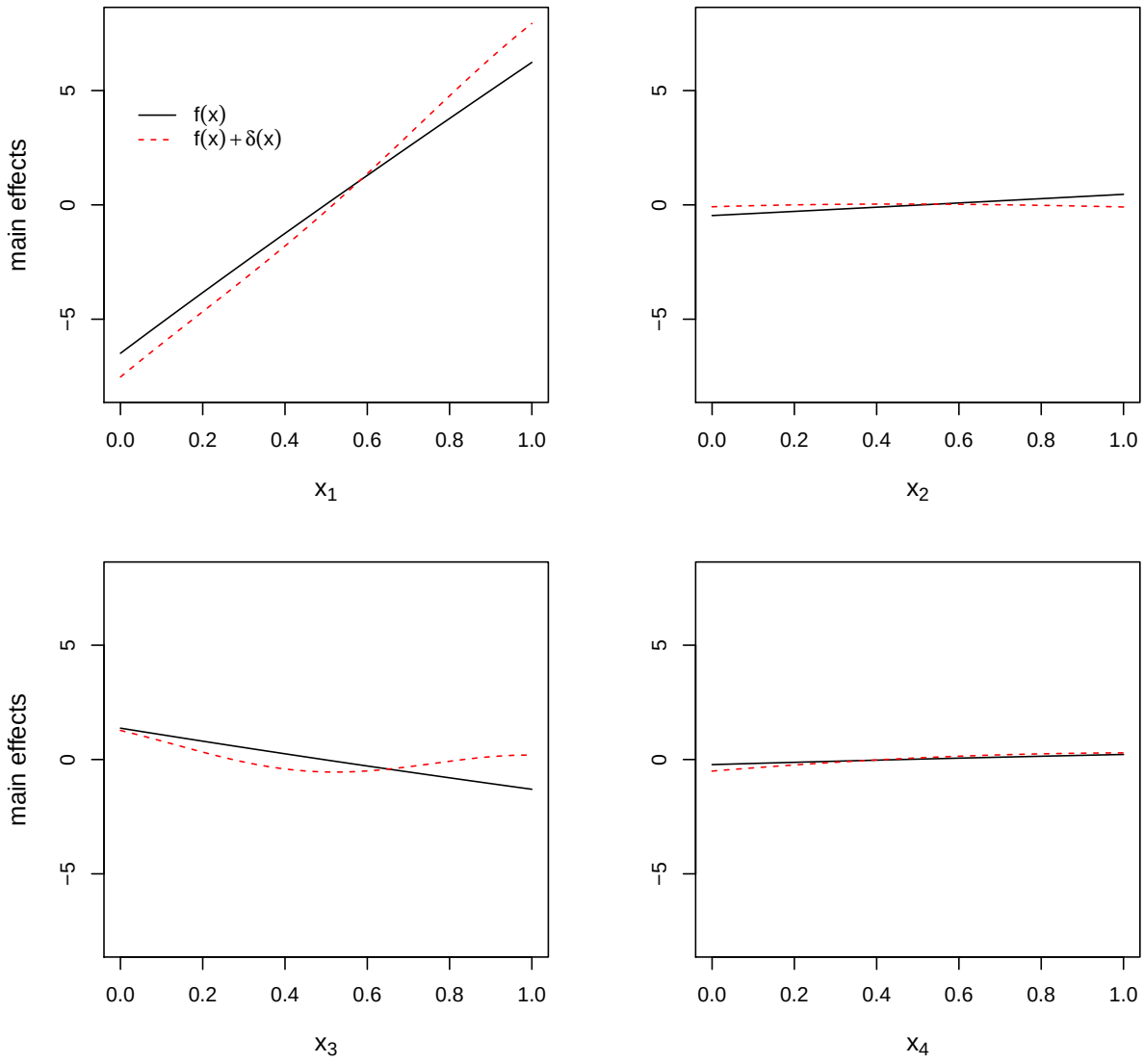

Figure 5: Plot of the main effects from the engineering model ($f(\boldsymbol{x})$) and the bias-corrected engineering model ($f(\boldsymbol{x}) + \delta(\boldsymbol{x})$).

achieved through a scale adjustment of the nominal depth of cut. Such variable-related adjustments are more informative than an overall adjustment. Thus, we choose (19) as the adjustment model for this problem.

Because $\hat{g}(\boldsymbol{x};\boldsymbol{\gamma})$ is a simple nonlinear regression model, we can easily obtain

$$\frac{\partial}{\partial\gamma_1}\hat{g}(\boldsymbol{x};\boldsymbol{\gamma}) = \frac{\hat{\beta}_1}{\gamma_1}\hat{g}(\boldsymbol{x};\boldsymbol{\gamma}), \quad \text{and} \quad \frac{\partial}{\partial\gamma_3}\hat{g}(\boldsymbol{x};\boldsymbol{\gamma}) = -\hat{\beta}_3 x_3 e^{-\hat{\beta}_4 x_4}\hat{g}(\boldsymbol{x};\boldsymbol{\gamma}).$$

For more complicated models, one can use numerical differentiation to obtain the

gradients, e.g., using the R package *numDeriv* (Gilbert 2012). Now the $\boldsymbol{G}_0$ matrix can be constructed by evaluating the gradients at $\boldsymbol{\gamma}_0 = (1, 1)'$. Assume a noninformative prior for σ_e^2: $p(\sigma_e^2) \propto 1/\sigma_e^2$. We generated an MCMC sample of size $m = 10,000$ using Metropolis algorithm. The posterior mean of the adjustment parameters is obtained as $\bar{\boldsymbol{\gamma}} = (1.17, .44)'$. Thus, we can see that the data suggests increasing the sensitivity of x_1 and decreasing the sensitivity of x_3. Decreasing the sensitivity of the laser power x_3 was a bit uncomfortable for the engineers because it means that the laser has less effect on reducing the cutting force. However, further investigation showed that the actual depth of cut was larger than the nominal depth of cut at higher laser powers due to the increased softening of the material. Thus by increasing the laser power we can cut more material, but it further increases the cutting force (because the force increases with x_1). This phenomenon was not captured in the engineering model and therefore, it makes perfect sense to reduce the sensitivity of laser power through a statistical adjustment. Thus, a plug-in estimate of the new engineering-statistical model is given by

$$\hat{g}(\boldsymbol{x}; \bar{\boldsymbol{\gamma}}) = 1.36(1.17 \times x_1)^{.89} \exp\left\{.0014x_2 - .0268(.44 \times x_3)e^{-.0034x_4}\right\}.$$

This is exactly the same as the statistically-adjusted engineering model obtained by Singh et al. (2011) through trial-and-error methods. It was validated using additional experiments and found to work well in practice.

Figure 6 shows the mean prediction and the approximate 95% prediction intervals computed using the MCMC sample for x_1 (by fixing $x_2 = 50, x_3 = 5$, and $x_4 = 200$) and x_3 (by fixing $x_1 = 20, x_2 = 50$, and $x_4 = 200$). For comparison, Kennedy and O'Hagan's model in (1) is fitted with $\rho = 1$ (as in Bayarri et al. 2007) and $h_0(\boldsymbol{x}) = 1$. The fitted model (denoted as KO adjustment model) is plotted in the same figure. We can see that the KO adjustment model and the proposed engineering-statistical model are in close agreement. The KO adjustment model is closer to the observed points, but its shape is quite different from the original engineering model. On the other hand, the proposed engineering-statistical model makes only minor changes to the engineering model and therefore, has better physical interpretations. For example, if

we reduce x_1 to 0, the KO adjustment model gives a force value 3.33, whereas the proposed engineering-statistical model agrees with the reality that there should be zero force when there is no cut.

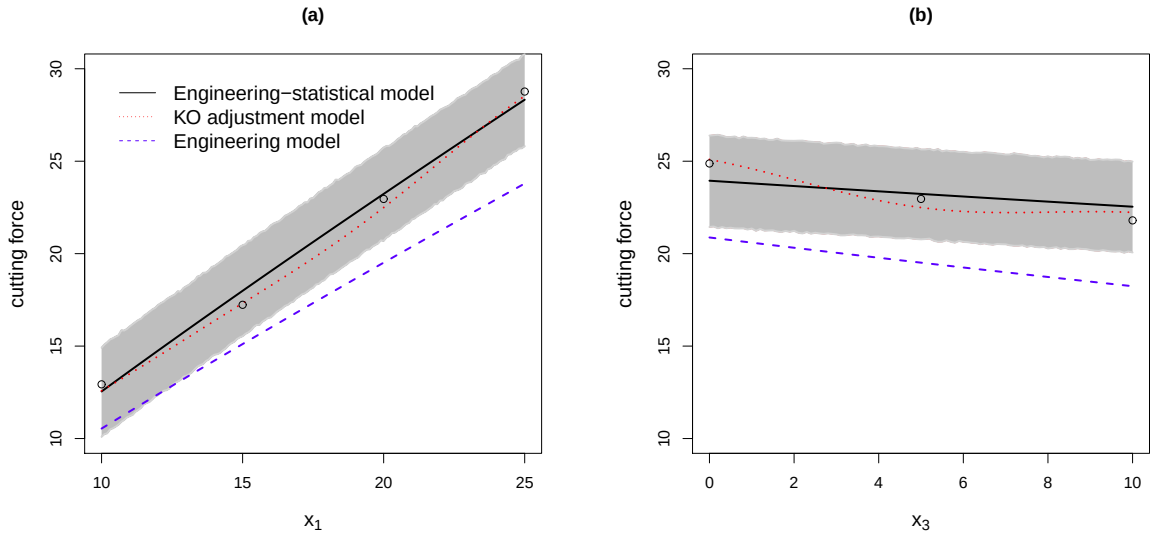

Figure 6: Predictions and 95% prediction intervals with respect to (a) x_1 (at $x_2 = 50, x_3 = 5$, and $x_4 = 200$) and (b) x_3 (at $x_1 = 20, x_2 = 50$, and $x_4 = 200$).

The predictions from the engineering-statistical model are plotted in Figure 7, which can be compared with Figure 2 to see the improvement. We can quantify the improvement using the following mean squared prediction errors, which we call model inadequacy (MI) measures. For the engineering model

$$MI = \frac{1}{n}\sum_{i=1}^{n}\{y_i - f(\boldsymbol{x}_i)\}^2 = 14.02,$$

whereas for the engineering-statistical model

$$MI = \frac{1}{n}\sum_{i=1}^{n}\{y_i - \bar{g}(\boldsymbol{x}_i)\}^2 = 1.41.$$

This shows about 90% improvement in the prediction accuracy, which is quite substantial.

The MI for the KO adjustment model is 0.21, indicating an even better fit to the data. However, this excellent fit is obtained through GP-based nonparametric modeling, which has a disadvantage. Suppose some modifications are made to the LAMM

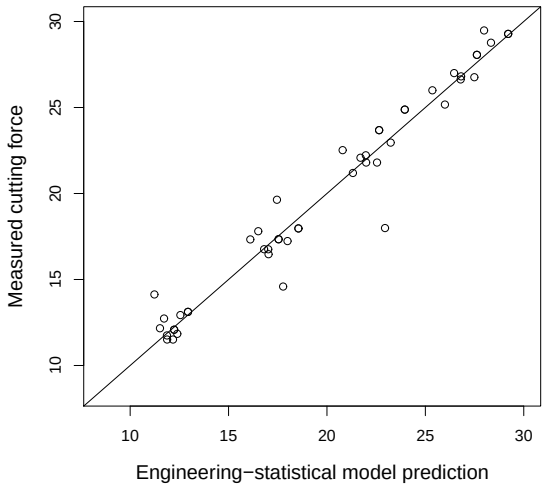

Figure 7: Plot of the cutting force predictions from the engineering-statistical model and the measured values.

process, such as changing the machine or material. Then, the whole experiment (with 48 runs) will have to be repeated to estimate the new KO adjustment model. On the other hand, only a few runs (say, 4 runs) are needed to estimate the two coefficients γ_1 and γ_3 in the new engineering-statistical model. This provides great savings for the investigator.

1.4.2 Fluidized-Bed Processing Example

Dewettinck, et. al. (1999) proposed a thermodynamic model to simulate the steady-state thermodynamic operation point of the fluidized-bed unit Glatt GPCG-1 in the top-spray configuration. To validate the engineering model, they conducted 28 runs of experiment for the GPCG-1 fluidized-bed process. There are six input variables: air relative humidity in the room (x_1), room temperature (x_2), inlet air temperature (x_3), flow rate of the solution (x_4), pressure of atomization air (x_5), and fluid velocity of fluidization air (x_6). The experiment output is the temperature of the outlet air.

Three versions of the model, Model I, Model II,and Model III, were developed and discussed in Dewettinck, et. al. (1999). In this chapter, we will use Model I as

the engineering model. Figure 8 plots the prediction of Model I and the measured temperature of outlet air in the fluidized-bed process. We can see that most of the values lie below the 45^o line indicating that the engineering model over-predicts the outlet air temperature. Thus, there is an obvious model discrepancy.

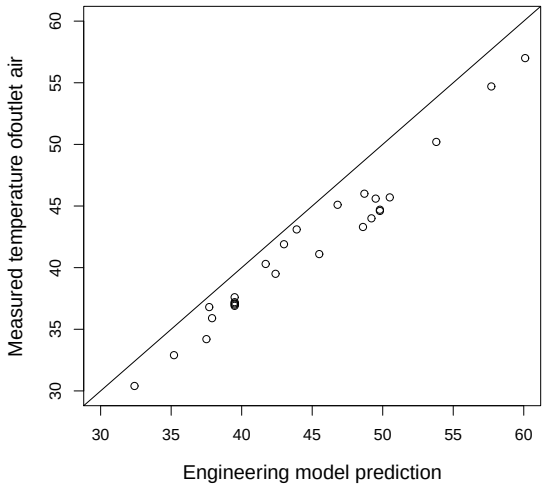

Figure 8: Plot of the prediction of outlet air temperature from the engineering model and the measured values.

Different from LAMM example, the cause of model discrepancy is already known. The deficiency of the engineering model is mostly because the heat losses are not taken into account. Since the cause of discrepancy is clear, it can be used to test if our proposed methodology can be used to obtain the right guidance of model adjustment. The heat losses consist of convective and radiative heat losses. According to the equations (46), (47), and (51) in Dewettinck, et. al. (1999), convective heat loss is influenced by inlet air temperature (x_3), room temperature (x_2), and pressure of atomization air (x_5), and radiative heat loss is mostly governed by x_2. Therefore, the identification of x_3, x_2, and x_5 is the key for this problem.

We first fit an ordinary kriging model to approximate the complex engineering model based on the 28-run simulations. Let $\boldsymbol{x} = (x_1, \cdots, x_6)'$ and denote the output of the engineering model by $\boldsymbol{y}_c$. The predictor of the ordinary kriging metamodel

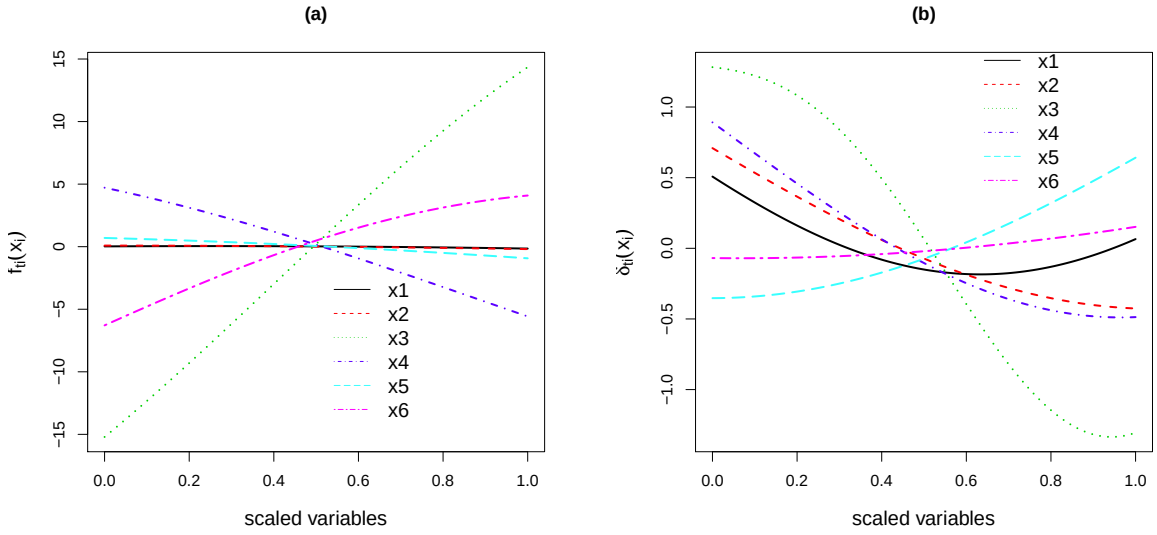

Figure 9: Main effects from the functional ANOVA decomposition of (a) $f_t(\boldsymbol{x})$ and (b) $\delta_t(\boldsymbol{x})$.

(Santner et. al, 2003) is

$$\hat{f}(\boldsymbol{x}) = \mu_0 + \boldsymbol{d}(\boldsymbol{x})\boldsymbol{D}^{-1}(\boldsymbol{y}_c - \mu_0\mathbf{1}), \tag{20}$$

where μ_0 is the mean, $\boldsymbol{D}$ is a 28×28 correlation matrix with ijth element $D(\boldsymbol{x}_i - \boldsymbol{x}_j) = \exp\{-\sum_{k=1}^{6} d_k(x_{ik} - x_{jk})^2\}$, and $\boldsymbol{d}(\boldsymbol{x}) = (D(\boldsymbol{x} - \boldsymbol{x}_1), \cdots, D(\boldsymbol{x} - \boldsymbol{x}_{28}))'$. The maximum likelihood estimators (Santner et. al, 2003) of μ_0 and $\boldsymbol{d}$ are $\mu_0 = 42.84$ and $\boldsymbol{d} = (0.063, 0.040, 0.367, 0.185, 0.040, 0.277)'$, respectively.

We then fit a GP model for the model discrepancy as described in Section 1.4.1. Following the procedure we have done for LAMM example, we obtain the correlation parameters $\hat{\boldsymbol{\theta}} = (0.615, 0.362, 1.950, 0.467, 0.307, 0.100)'$ and the variance parameters $\hat{\sigma}^2 = 0.19$ and $\hat{\tau}^2 = 4.66$. The main effects of $\hat{f}(\boldsymbol{x})$ and discrepancy function $\hat{\delta}(\boldsymbol{x})$ can be computed explicitly. The two main effects are plotted in Figure 9. For the main effects of $f_t(\boldsymbol{x})$, we can see that the inlet air temperature (x_3) has the largest effect on the temperature of outlet air. In Figure 9 (b), we can find that the inlet air temperature (x_3) is also the main contributor of the model discrepancy. The contributions of the other factors to model and discrepancy are more clear in Figure 10, which plots the sensitivity indices of main effects and two-factor interactions .

We can see that the largest sensitivity index is due to the main effect of x_3. The next largest three main effect sensitivity indices are of x_4, x_2 and x_5, respectively. The sensitivity indices of x_1, x_6 and two-factor interactions are small and therefore, a main effects analysis is reasonable for this problem.

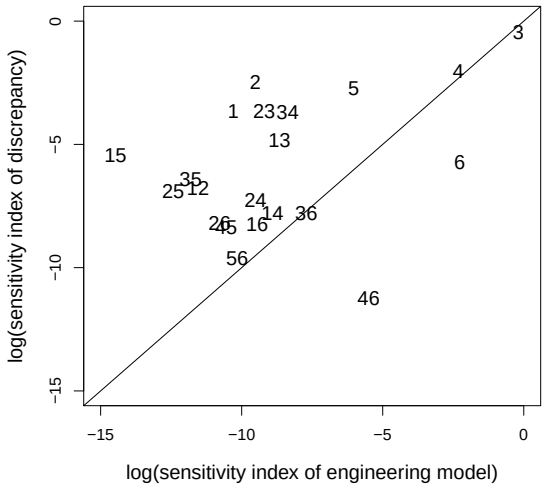

Figure 10: Plot of the log-sensitivity indices of engineering model and discrepancy in fluidized-bed processing example.

From the plots of main effects and sensitivity indices, we can conclude that the discrepancy is mostly related to inlet air temperature (x_3). It could also be relevant to flow rate of the solution (x_4), room temperature (x_2) and pressure of atomization air (x_5). Now, we successfully identify x_3, x_2 and x_5, which are influential to the heat losses. Moreover, from our analysis, the convective heat loss (related to x_3, x_2 and x_5) could be the major part of the heat loss in the fluidized-bed process, while the radiative heat loss (governed by x_2) could be a minor part. The identification of x_3, x_2 and x_5 will provide a very useful guidance for engineers to consider adding heat losses as potential improvement for the engineering model. Regarding x_4, it may account for some other deficiency in the engineering model.

The next step is to postulate an adjustment model. From a pure engineering point of view, this has been done in Dewettinck, et. al. (1999). The engineering

model was developed by adding corresponding physical equations which quantify the heat losses in the process. Here, we also want to try improving the engineering model using a statistical adjustment and see how it works compared with the pure engineering-driven adjustment. From the sensitivity analysis, we already found x_3 is most significant among the six input variables. Therefore, the adjustment related to x_3 is required. The adjustment of x_4, x_2, and x_5 may also be necessary. The sensitivity index of x_4 is larger than the sensitivity indices of x_2 and x_5, while x_2 and x_5 are very close.

To decide which type of adjustment we should apply, we plot the main effects of the engineering model plus the discrepancy function along with the main effects of the engineering model in Figure 11. From Figure 11, we can see that the data is to slightly tilt the main effects of x_2, x_3, x_4 and x_5, and the tiltings are approximately linear. Then a scale adjustment model might work. In addition, from Figure 10, we can find that x_3 and x_4 have significant effects on both model and discrepancy, while x_2 and x_5 are only significant on discrepancy. Then the adjustment of x_2 or x_5 inside model can hardly help. Instead, we choose to account for the adjustment of x_2 and x_5 by adding the terms including them to the model. We propose four engineering-statistical adjustment models, Model#1, Model#2, Model #3, and Model#4 as below.

Model #1:

$$\hat{g}_t(\boldsymbol{x};\boldsymbol{\gamma}) = \hat{f}_t(x_1, x_2, \gamma_3 x_3, x_4, x_5, x_6)$$

Model #2:

$$\hat{g}_t(\boldsymbol{x};\boldsymbol{\gamma}) = \hat{f}_t(x_1, x_2, \gamma_3 x_3, \gamma_4 x_4, x_5, x_6)$$

Model #3:

$$\hat{g}_t(\boldsymbol{x};\boldsymbol{\gamma}) = \hat{f}_t(x_1, x_2, \gamma_3 x_3, \gamma_4 x_4, x_5, x_6) + \gamma_2 x_2$$

Model #4:

$$\hat{g}_t(\boldsymbol{x};\boldsymbol{\gamma}) = \hat{f}_t(x_1, x_2, \gamma_3 x_3, \gamma_4 x_4, x_5, x_6) + \gamma_2 x_2 + \gamma_5 x_5.$$

In Model #1, we only consider the adjustment of x_3. In Model #2, we consider the adjustment of x_3 and x_4. The adjustment of x_3, x_4, x_2 and x_5 are all considered

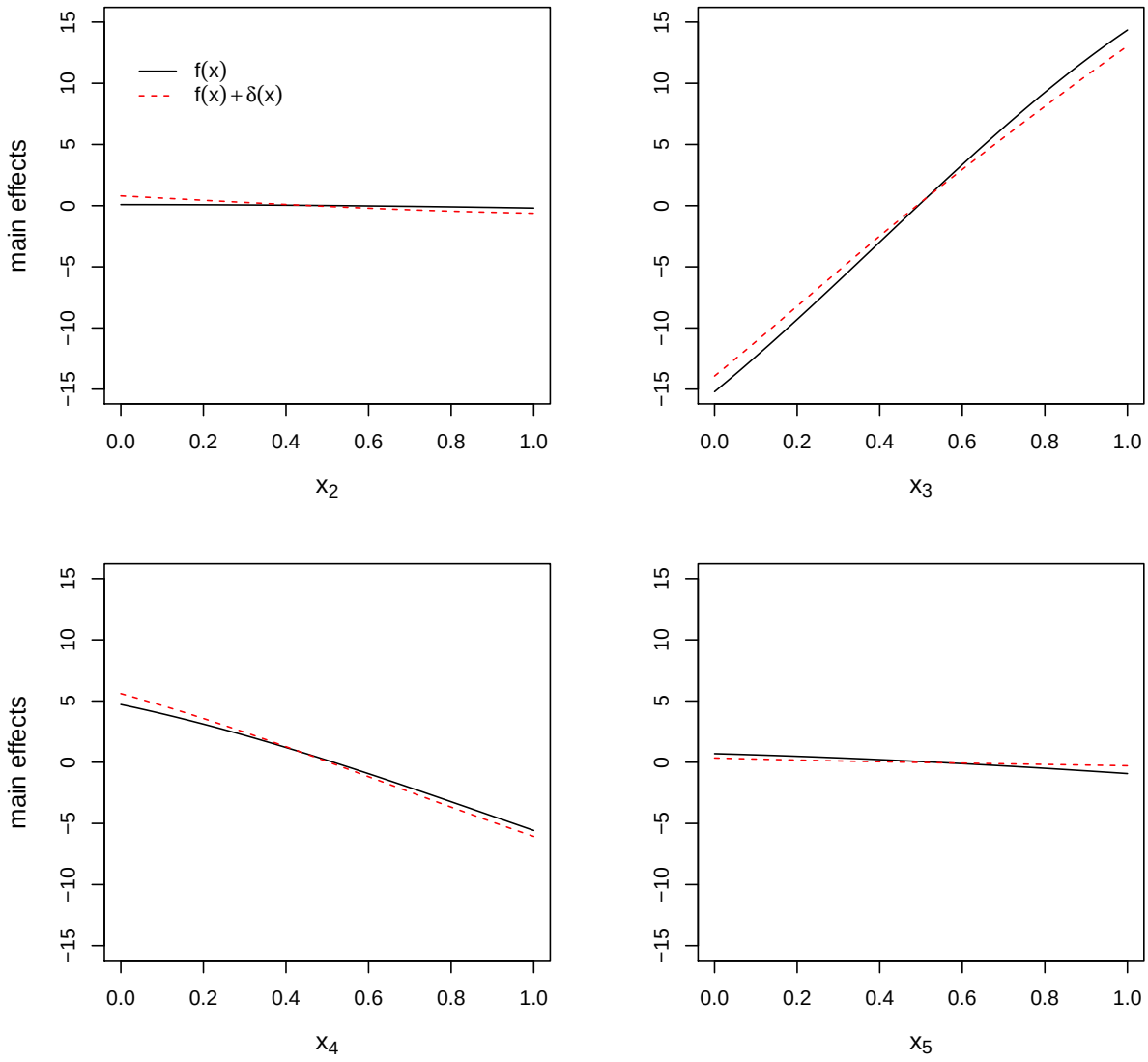

Figure 11: Plot of the main effects from the engineering model ($f_t(\boldsymbol{x})$) and the bias-corrected engineering model ($f_t(\boldsymbol{x}) + \delta(\boldsymbol{x})$).

in Model #3. The $\boldsymbol{\gamma}_0$ for the three models are 1, $(1,1)'$, $(0,1,1)$, and $(0,1,1,0)'$, respectively. Using MCMC method, we get the posterior mean of the adjustment parameters. $\bar{\boldsymbol{\gamma}}$ for the three models are 0.937, $(0.970, 1.11)'$, $(-0.001, 0.970, 1.112)'$, and $(-0.073, 0.961, 1.098, 0.737)'$, respectively. We can see that the estimates suggest that decreasing the sensitivity of x_2 and x_3 and increasing the sensitivity of x_4 and x_5. We can quantify the model improvement using the MI again. For the engineering

model

$$MI = \frac{1}{n}\sum_{i=1}^{n}\{y_{ti} - f_t(\boldsymbol{x}_i)\}^2 = 10.59,$$

For the engineering-statistical models, we compute their MI's using

$$MI = \frac{1}{n}\sum_{i=1}^{n}\{y_{ti} - \bar{g}_t(\boldsymbol{x}_i)\}^2.$$

In Table 1, we compare the MI's of the engineering model and the three engineering-statistical models. We can see that even only adjusting x_3, we can achieve about 92% improvement in prediction accuracy, which is quite substantial. If we consider the Model #3 with adjustment on x_2, x_3, $x_4$and x_5, we can have an improvement in prediction about 95%. The predictions from Model #1, are plotted in Figure 12, which can be compared with Figure 8 to see the improvement.

Model	MI
Engineering Model	10.59
Model #1 (x_3)	0.875
Model #2 (x_3, x_4)	0.698
Model #3 (x_2, x_3, x_4)	0.698
Model #4 (x_2, x_3, x_4, x_5)	0.521

Table 1: MI's of engineering model and engineering-statistical models

An interesting phenomena is that the improvement achieved by statistical-engineering model is almost the same as the one gained by the optimized model ($MI = 0.69$) in Dewettinck, et. al. (1999). It is amazing that a simple statistical-engineering adjustment can play an equally effective role that a pure engineering adjustment does, whereas the former takes much less effort and computation than the latter does. The proposed statistical adjustment is also much simpler than the one discussed in Reese et al. (2004).

1.5 Model Refinement

The adjustment models described in Section 1.2.2 are simple, but are not as flexible as the GP modeling approach for the discrepancy term. We have sacrificed this flexibility to gain on the physical interpretability. However, the ultimate objective is

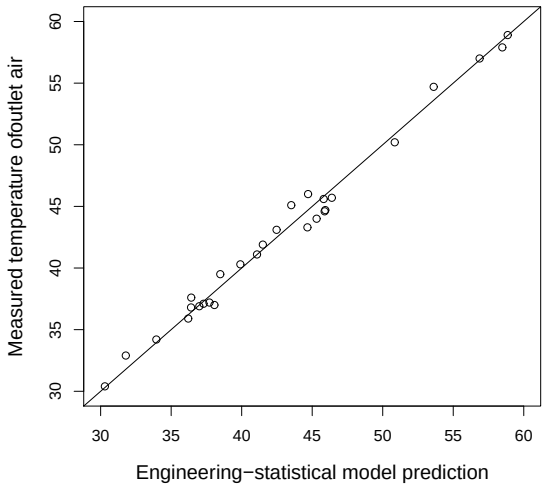

Figure 12: Plot of the prediction of outlet air temperature from the engineering-statistical model (Model #1) and the measured values.

to construct a predictive model and therefore, follow-up strategies should be used if the engineering-statistical model turned out to be inadequate in terms of prediction performance. In the laser-assisted mechanical micromachining example, we were able to obtain significant improvement in prediction using a simple scale adjustment model. But we need not be that fortunate always. This can happen, for example, if we encounter a situation where some variables have significant effects on the discrepancy but almost no effect on the engineering model or there is a significant higher order interaction effect on the discrepancy which cannot be captured by the adjustment models mentioned in Section 1.2.2. In such cases, other adjustment models should be considered. Our hope is that the engineers and statisticians working in that system will be able to postulate a better adjustment model that is physically interpretable. However, this can be a difficult task especially in high dimensional problems. Therefore, our objective here is to present a follow-up strategy that can be used when we encounter such a situation.

The basic idea is to treat the adjusted model $g(\boldsymbol{x};\boldsymbol{\alpha})$ as the "new" engineering model (see Figure 1). This provides enough flexibility for the investigator to refine

the model using an appropriate modeling strategy. For example, the investigator may choose to use a pure engineering approach to fix the inadequacy in $g(\boldsymbol{x};\boldsymbol{\alpha})$ or a pure statistical approach to capture the discrepancy using a statistical model as in Kennedy and O'Hagan (2001) or even use an engineering-driven statistical approach as described in this chapter. This may require new data collection and/or bringing in new variables that were missed in the initial study.

Consider, for example, a pure statistical approach. We will now fit the following model to the data:

$$Y = g(\boldsymbol{x};\boldsymbol{\alpha}) + e(\boldsymbol{x}) + \epsilon,$$

where $e(\boldsymbol{x}) \sim GP(0, \sigma_e^2 R_e(\cdot))$ is the new discrepancy function which is modeled using a smooth GP (instead of the white noise error in (12)). Note that different from the model presented in Section 1.2.1, $g(\boldsymbol{x};\boldsymbol{\alpha})$ becomes the prior mean of the unknown true function instead of the original engineering model $f(\boldsymbol{x};\boldsymbol{\eta})$. An interesting special case is when $g(\boldsymbol{x};\boldsymbol{\alpha}) = \gamma_0 + \gamma_1 f(\boldsymbol{x};\boldsymbol{\eta})$, which reduces to the Kennedy and O'Hagan's model in (1) and (2) with $\mu_0 = \gamma_0$, $\rho = \gamma_1$, $h_0(\boldsymbol{x}) = 1$, and the rest of the μ's equal to 0. This is the reason why we started with a simplified model in Section 1.2.1.

If we need to follow-up with a pure statistical approach, one may wonder why not use the statistical approach at the first place and always. The basic philosophy here is to fix the engineering model as much as possible using the engineering-driven statistical approach and then to use a pure statistical approach to capture the remaining discrepancy, if needed.

1.6 Conclusions

Some engineering systems are so complex that it is almost impossible to capture all the details for accurately modeling the system. Many assumptions will have to be made for tractable mathematical modeling. A multitude of such assumptions introduce errors in the model predictions. The engineering approach to find where the assumptions went wrong and fix them is time consuming and often infeasible in industrial practice. The statistical approach offers a quick solution, that is to correct

the observed discrepancy using a statistical model estimated from the real data. Although this is found to be very effective in practice, it does not possess the physical interpretations that the engineering model has. The statistical approach treats the engineering model as a black box and focuses only on the observed discrepancy. On the other hand, this chapter proposes an approach to open up the black box and make statistical adjustments *inside* the engineering model. This approach has a tremendous value in that it offers better physical interpretations. Moreover, such models are found to do better in extrapolation and can adapt easily to small modifications in the system.

The main ideas in our approach are very simple. It uses some main effects analysis and graphical plots to identify the potential causes of discrepancy. We hope that such a simple approach will appeal to the practitioners and engineers working in industries. More sophisticated analysis that makes use of higher order effects analysis can be developed in the future. The adjustment models that we proposed such as the scale adjustment model are also very simple. We found that it works well in many real examples. However, they should not to be used as a general recipe for statistical adjustment. Instead, they should be treated as some guidelines for developing adjustment models that are suitable for the problem in hand.

Although not discussed in detail, it should be clear that the experimental designs for the computer experiment and the physical experiment play a crucial role in the success of the engineering-driven statistical adjustment methodology. A common strategy that one could adopt is to use a space-filling design for the computer experiments (see Santner et al. 2003 and Fang et al. 2006) and a fractional factorial design for the physical experiments (see Wu and Hamada 2009). Moreover, for a model-free estimation of the discrepancy, it is desirable to have the design for the physical experiment nested within the design for the computer experiment (Qian, Tang, and Wu 2009). In addition, since we are interested in the efficient estimation of the main effects, orthogonality of the space-filling designs is also desirable (see, e.g., Lin et al. 2010). Thus, there are many aspects that should be tied together for the efficient generation of the experimental designs. Much remains to be done in this area.

In this chapter, we have focused our attention to a single output case, while most engineering systems have multiple outputs. For example, in the laser-assisted mechanical micro-machining example, the engineering model produces two force outputs: cutting force and thrust force, in which we have analyzed only the cutting force in Section 1.4. Of course, the same analysis can be repeated on the thrust force as well, but more insights can be obtained through a simultaneous analysis of the two forces. We leave this interesting topic of statistical adjustment with multiple outputs for future research.

CHAPTER II

CALIBRATION WITH RANDOM EFFECTS: AN APPLICATION IN CARDIAC CELL MODELING

2.1 Introduction

Many of the computer models contain unknown parameters, known as *calibration parameters*, which need to be estimated from real data before making a prediction. While this can be formulated as a nonlinear regression problem, complications arise at many different levels making it difficult to solve using standard regression techniques. First, the underlying mathematical model which is implemented as a computer code is often a simplified version of the real system. Many of the simplifying assumptions used in deriving the mathematical model may not hold true in practice leading to biased predictions. Second, the computer model is computationally expensive to evaluate and therefore, a direct estimation using a nonlinear least squares method is almost impossible. The latter can be addressed by approximating the computer model using an easy-to-evaluate metamodel (Sacks et al. 1989), whereas the former can be addressed by introducing a bias function in the statistical model to capture the bias in the computer model (Kennedy and O'Hagan 2001). However, this is not a trivial problem because the approximation of the computer model, the estimation of the bias function, and the estimation of calibration parameters are all inter-connected. Different approaches to this challenging statistical problem have been proposed in the literature: Higdon et al. (2004), Reese et al. (2004), Bayarri et al. (2007), Qian and Wu (2008), Joseph and Melkote (2009), and Pratola et al. (2013). But as described below, the problem that we encountered brought new challenges that were not addressed in these earlier works.

This chapter studies the calibration of a computer model that describes the potassium (K^+) currents in a cardiac cell. Real measurements of the electrical signals over

time for different settings of the voltages were obtained through an experiment on mouse cells, which can be used for calibrating the computer model. A major challenge in this problem is the high dimensionality of the model. There are a total of 24 calibration parameters that need to be estimated. In the traditional approach (see, e.g., Santner et al. 2003), first a computer experiment needs to be performed by varying the voltage and the calibration parameters. This is not a small problem because we would need 2^{24} (approximately 16.8 million) points even to fill up the corner points of a 24-dimensional hypercube. Much smaller sample sizes are recommended in the computer experiments literature such as $10d$, where d is the number of dimensions. However, our initial attempts to approximate the K^+ currents using kriging (Sacks et al. 1989) did not produce satisfactory results even with the use of about $100d$ points. We were forced to look for other alternatives for approximating the computer model.

The mathematical model for the K^+ current is a system of ordinary differential equations (see Appendix). There is no explicit solution for these equations and they are solved using some iterative numerical methods. Since our initial attempts to apply traditional methods failed, we started to look more closely into the mathematical model. We found out that parts of the differential equations can be solved and through a statistical model they can be connected together, producing an excellent approximation to the final output of the computer model. Although our approach may not be generalizable to other problems, the success of our study reinforces the often neglected idea that whenever possible, system knowledge should be used to supplement data-driven methods, especially in solving complex engineering problems.

Although finally we were able to successfully approximate the computer model, even more challenges were waiting for us at the calibration stage. The bias of the computer model was a minor issue in our problem because of the flexibility provided by the large number of calibration parameters. In fact, we did not even use a bias function in our statistical model as in Kennedy and O'Hagan (2001) and others. The main problem was that the measured currents vary quite a lot between each cell. As with all biological systems, this cell-to-cell variation is expected as each cell is different in terms of its size, shape, capacitance, conductance, and other characteristics. Thus,

estimated calibration parameters were found to vary drastically with each cell. One way to address this problem is to model the calibration parameters as random effects. We had measurements from five mouse cells. This implies we have a total of $24 \times 5 = 120$ random effects. Marginalizing the likelihood over these random effects and estimating the mean parameters and variance components from this nonlinear mixed-effects model is not an easy problem. In this chapter we propose a novel method to identify the significant random effects and thus, drastically reducing the dimension of the problem.

The article is organized as follows. In Section 2.2, we present the computer modeling of the cardiac cell currents and the voltage clamp experiments on the mouse cell. In Section 2.3, we describe our metamodeling approach used for approximating the computer model. In Section 2.4, we discuss the calibration of the computer model using the data from the voltage clamp experiment. Some concluding remarks and future research directions are given in Section 2.5.

2.2 Computer Model and Data

The action potential of a heart cell, the fast event where electrical impulses pass through a cell's membrane, is critical to the correct function of the entire organ. Because of this activity's importance, mathematical models of the electrophysiological process have been developed to explain the cardiac action potential. The first quantitative model of action potential was formulated by Hodgkin and Huxley (1952), which models the major current activity with a group of nonlinear ordinary differential equations. Models of cardiac action potential were developed in the following decades by incorporating the dynamic changes of the intracellular concentration of sodium and potassium: Rasmusson et al. (1990); Courtemanche et al. (1998); Noble et al. (1998); Hund and Rudy, (2004); Bondarenko et al. (2004). Rudy and Silva (2006) conduct a thorough review of these types of models. This work focuses on the potassium channel model proposed by Bondarenko et al. because their model was designed for cells similar to our case study. The model is based on the principle that ion channels open and close depending on the applied potential across the cell's

membrane. The dynamics of this model are similar to a collection of ion channels behaving according to a Markov model, therefore we refer to this as a Markov ordinary differential equation model. The process of ion channels opening and closing is referred to as *gating*. The amount of current through an open channel is controlled by *conductance* parameters. Conductance parameters influence the amplitudes of response current and gating parameters generally influence the shape of the response current.

Our experiment targets the current through channels that transfer potassium ions. There are seven types of channels in the Markov ordinary differential equation model proposed by Bondarenko et al. that transfer potential via potassium ions. Since rapid delayed rectifier, slow delayed rectifier and time-independent potassium current are very small, they are ignored in this work. The transient outward potassium current is also not considered because these channels conductances' equal zero in an apical cell (Xu. et al. 1999 and Bondarenko et al. 2004). Thus, we focus on the three major potassium currents: transient outward potassium current ($I_{Kto,f}$), ultrarapidly activating delayed rectifier potassium current (I_{Kur}), and noninactivating steady-state potassium current (I_{Kss}). The Markov ordinary differential equations governing these three currents are stated in the Appendix. The membrane potential is denoted as v and the time is denoted as t. We have carefully parameterized the ordinary differential equations given in Bondarenko et al. to avoid identifiability issues. The model contains twenty-four parameters, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{24})'$, that can be partitioned into three conductance parameters (η_1, η_2, η_3) and twenty-one gating parameters $(\eta_4, \ldots, \eta_{24})$. The solution to these ordinary differential equations given by the solver ode45 in Matlab represents our computer model (Du et al. 2013). It takes about 100 seconds to obtain one output from this computer model running on a 3.5 GHz desktop computer.

Our data consists of the potassium currents of five distinct apical cells from ST3Gal4-\- mice. In ST3Gal4-\- mice, a uniformly expressed cardiac glycogene responsible for glycoprotein sialylation, the sialyltransferase beta-galactoside alpha-2,3-sialyltransferase 4 (ST3Gal4), was deleted. See Ednie et al.(2013) for details on cell

preparation and electrophysiology. In the voltage clamp experimental protocol, depolarizing pulses of length 4.5 seconds were applied to the five ST3Gal4-\- apical cells with membrane potentials ranging from -50 to 50 mV in 10 mV increments from the holding potential of -70 mV. The intracellular and extracellular solutions were designed such that the only available ion to pass through the channels was potassium. The flow of ions across the cell is then recorded by an ammeter for the duration of the pulse.

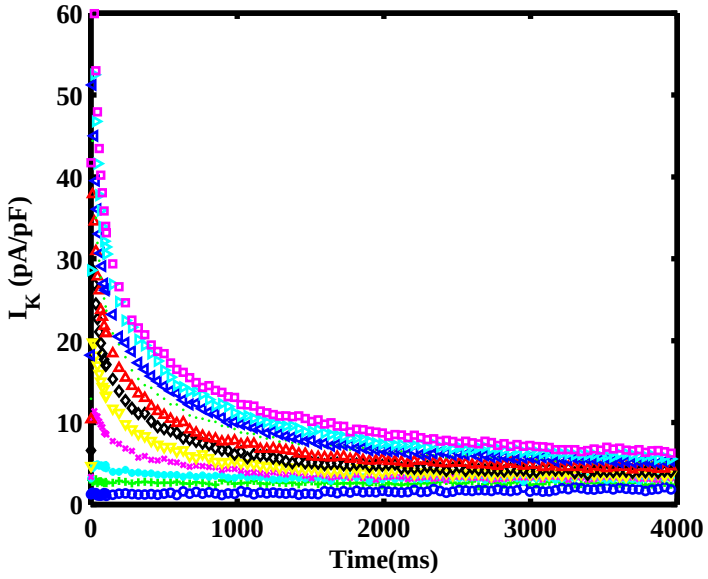

Figure 13: Potassium current recorded from a ST3Gal4-\- apical cell using a whole-cell voltage clamp protocol with depolarizing membrane potential between -50 mV to + 50 mV in 10 mV increments from a holding membrane potential of -70 mV.

The potassium currents for one of the five cells are shown in Figure 13. The current increases quickly, reaches peak value, and then decreases slowly. The amplitude and the rates of rise and decay are dependent on the membrane potential. Figure 14 shows potassium currents from five distinct apical cells at a membrane potential equal to +50 mV. We can see that the decay phases of potassium currents are cell dependent. The respective amplitudes of the current densities also appears to vary among the five cells.

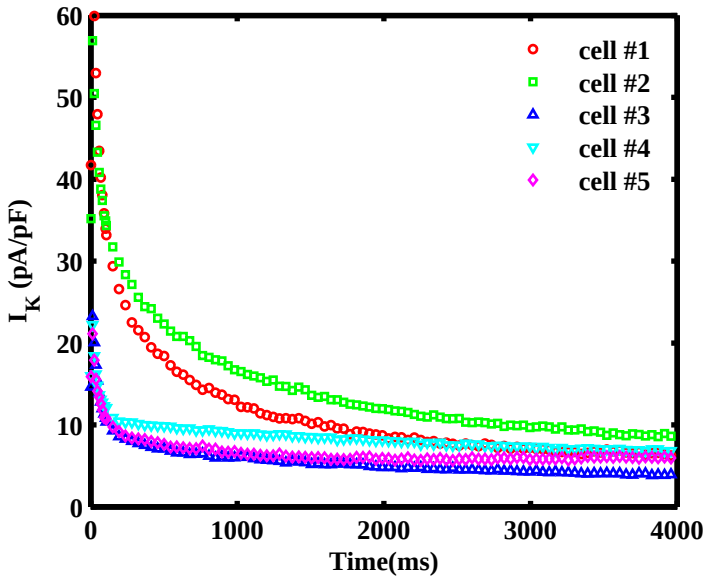

Figure 14: Potassium currents recorded from five distinct ST3Gal4-\- apical cells using a whole-cell voltage clamp protocol with depolarizing membrane potential of +50 mV from a holding membrane potential of -70 mV.

2.3 Physics-driven Metamodeling

At the beginning of this study, we tried to approximate the computer model using methods from design and analysis of computer experiments (Santner et al. 2003). We first started with a 240-run maximin Latin hypercube design (MmLHD) for the 24 calibration parameters. Then we ran the computer model for these 240 settings at each of the eleven voltages used in the voltage clamp experiment. This produced $240 \times 11 = 2640$ functional outputs of current over time. One hundred equally spaced time points were chosen producing 264,000 data points. A kriging model with Gaussian correlation function was fitted to this functional data using a Kronecker product simplification of the correlation matrix (Williams et al. 2006). Unfortunately, the prediction errors at some randomly observed settings of the calibration parameters were found to be too large. We increased the size of the MmLHD to 2000 runs (it took about seven days to generate the MmLHD in JMP in our 3.5 GHz desktop computer) and repeated the above procedure. Although the fitting was improved, still large discrepancies were noted, especially when predicting at locations far from the design points. Increasing the run size further is an option to improve the accuracy, but

the optimal design generation and fitting of the kriging model will be even more time consuming. Moreover, the cost of prediction also increases with the number of points, which is a serious issue because we need to use the fitted kriging model repeatedly within an optimization algorithm used for calibration.

Because of the difficulty with the traditional framework for computer experiments, we have decided to try a different approach. There are seven ODEs in the 20 equations of the currents given in the Appendix. They can be grouped into two levels. The high level ODE (A-1) describes the relationship between intracellular concentration $[\mathrm{K}^+]_i$ and current components as a function of time. There are six lower level ODEs: (A-4)-(A-5), (A-11)-(A-12), and (A-17)-(A-18). These ODEs define the six gating variables $a_{to,f}, i_{to,f}, a_{ur}, i_{ur}, a_{Kss}$ and i_{Kss}, which are functions of the transfer rate coefficients (α_a, β_a, a_{ss}, i_{ss}, $\tau_{ta,s}$, $\tau_{ti,s}$, τ_{aur}, τ_{iur}, and τ_{Kss}). Transfer rate coefficients are dependent on the membrane potential and the 21 gating parameters $\eta_4, \cdots, \eta_{24}$. The two levels of ODEs are linked by the equations (A-2), (A-3), (A-10), and (A-17).

Our aim is to obtain the total current $I_{K,sum}(t) = I_{Kto,f}(t) + I_{Kss}(t) + I_{Kur}(t)$. There is no explicit solution for this from the above system of ODEs, but we noticed that each of the six ODEs in the lower levels can be solved explicitly as a function of time. Thus, if we can get an expression for $E_k(t)$, then we can explicitly compute $I_{K,sum}(t)$. However, this is not possible because $E_k(t)$ depends on all the current components through the two equations (A-1) and (A-3). Here we can do an approximation. Examination of the plots in (13) shows that $I_{K,sum}(t)$ can be approximated by (neglecting the rapid rise of current in the beginning)

$$I_{K,sum}(t) \approx b_0 + b_1 \exp(-b_2 t). \tag{21}$$

Under this approximation, $[K^+]_i(t)$ can be solved explicitly from (A-1) as

$$\begin{aligned} [K^+]_i(t) &= [K^+]_{i,0} - \frac{A_{cap}C_m}{V_{myo}F} \int_0^t I_{K,sum}(t)\, dt \\ &\approx [K^+]_{i,0} - \frac{A_{cap}C_m}{V_{myo}F} \left[b_0 t - \frac{b_1}{b_2} \exp(-b_2 t) + \frac{b_1}{b_2} \right], \end{aligned} \tag{22}$$

where $[K^+]_{i,0}$ is the initial intracellular concentration. Substituting this in (A-3) gives us the desired expression for $E_k(t)$. Now the only remaining part is to obtain b_0, b_1,

and b_2. For quick computation, we let $\hat{b}_0 = I_{K,sum}(T)$, where T is the largest observed time point. Now we fit $\log I_{K,sum}(t) - \hat{b}_0 = \log b_1 - b_2 t$ using least squares to obtain $\hat{b}_1$ and $\hat{b}_2$. These can be done only if $I_{K,sum}(t)$ is known, which is actually unknown. Hence we solve it iteratively by initializing $E_K(t) \approx E_{K,0} = RT/F \ln([K^+]_o/[K^+]_{i,0})$ and obtaining an initial approximation for $I_{K,sum}(t)$. In fact, we found that the solution after one iteration itself is very accurate. We call this approximate model as physics-driven metamodel. It is given by

$$I_{K,sum}(t) = (\eta_1 a_{to,f}^3 i_{to,f} + \eta_2 a_{ur} i_{ur} + \eta_3 a_{Kss} i_{Kss}) \left[v - \frac{RT}{F} \ln \left(\frac{[K^+]_o}{[K^+]_i(t)} \right) \right], \quad (23)$$

where $[K^+]_i(t)$ is computed from (22) and

$$\begin{aligned}
a_{to,f} &= \frac{\alpha_a}{\beta_a + \alpha_a} + (a_{to,f0} - \frac{\alpha_a}{\beta_a + \alpha_a}) \exp\left[-(\beta_a + \alpha_a)t\right], \\
i_{to,f} &= \frac{\alpha_i}{\beta_i + \alpha_i} + (i_{to,f0} - \frac{\alpha_i}{\beta_i + \alpha_i}) \exp\left[-(\beta_i + \alpha_i)t\right], \\
a_{ur} &= a_{ss} + (a_{ur0} - a_{ss}) \exp\left(-\frac{t}{\tau_{aur}}\right), \\
i_{ur} &= i_{ss} + (i_{ur0} - i_{ss}) \exp\left(-\frac{t}{\tau_{iur}}\right), \\
a_{Kss} &= a_{ss} + (a_{Kss0} - a_{ss}) \exp\left(-\frac{t}{\tau_{Kss}}\right), \\
i_{Kss} &= i_{Kss0}.
\end{aligned}$$

The initial values $a_{to,f0}$, $i_{to,f0}$, a_{ur0}, i_{ur0}, a_{Kss0} and i_{Kss0} are given in the Appendix.

The predictions of $I_{Ksum}(t)$ from the kriging model, physics-driven metamodel and the Markov model are compared in Figure 15 for a random set of the calibration parameters. We can see that obvious discrepancies are between the the kriging model and the Markov model in the upper two voltage levels, whereas the the physics-driven metamodel almost overlaps the Markov model for all voltages throughout time span. Figure 16 compares E_{K0}, the one-step updated $E_K(t)$ in the physics-driven metamodel, and the true $E_K(t)$ in the Markov model. We can see that the difference between true $E_K(t)$ and E_{K0} increases with time, whereas the one-step updated $E_K(t)$ is very close to the true $E_K(t)$. This explains why the prediction of $I_{Ksum}(t)$ from the physics-driven metamodel agrees very well with the Markov model after one iteration.

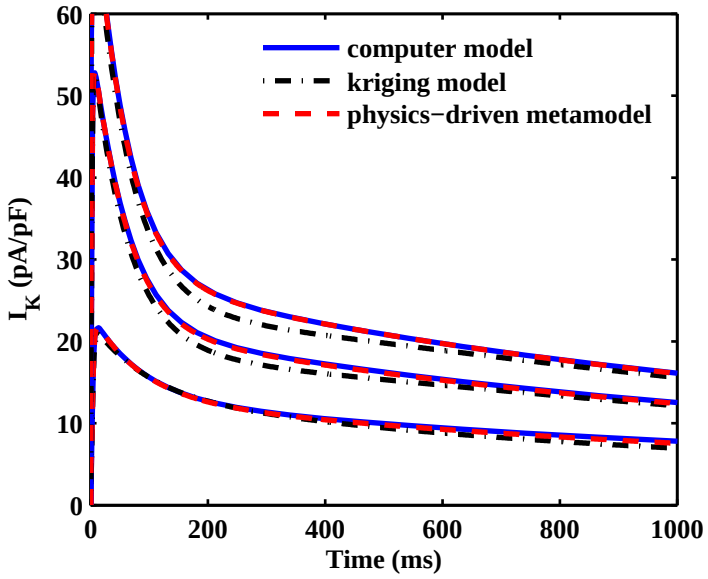

Figure 15: Predictions of $I_{Ksum}(t)$ from the kriging model, physics-driven model and computer model.

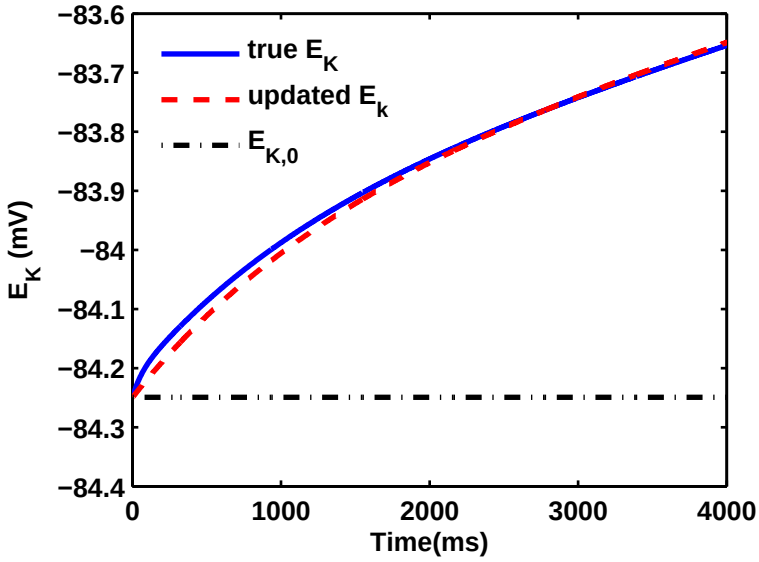

Figure 16: E_{K0} (black dotted dash line), updated $E_K(t)$ (red dashed line) and true $E_K(t)$(blue dash line)

2.4 Calibration

2.4.1 Initial Calibration Model

Let y_{ijk} be the measured potassium current at time t_i and voltage v_j of the kth cell with $i = 1, \ldots, n$, $j = 1, \ldots, m$, and $k = 1, \ldots, K$, where $n = 100$, $m = 11$, and $K = 5$. We have

$$y_{ijk} = f(t_i, v_j; \boldsymbol{\eta}) + \epsilon_{ijk}, \tag{24}$$

where $f(t_i, v_j; \boldsymbol{\eta})$ is the true model, $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_{24})'$ are the 24 calibration parameters, and $\epsilon_{ijk} \sim^{iid} N(0, \sigma_0^2)$ is the measurement error. To begin with we assume that the computer model is an accurate representation of reality and use the computer model in place of $f(t_i, v_j; \boldsymbol{\eta})$. Furthermore, since the metamodel developed in the previous section is quite accurate, we use the metamodel to replace the expensive computer model.

Since all conductances and gating parameters are positive, we transform them to $\lambda_l = \log(\eta_l)$ for $l = 1, \cdots, 24$. Using Inverse-Gamma prior for σ_0^2 and normal prior for $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_{24})'$, we have the following Bayesian model:

$$\begin{aligned} y_{ijk}|\boldsymbol{\lambda}, \sigma_0^2 &\sim^{ind} N(f_{ij}(\boldsymbol{\lambda}), \sigma_0^2), \\ \lambda_l &\sim^{ind} N(\mu_l, \tau_l^2), l = 1, \cdots, 24 \\ \sigma_0^2 &\sim IG(a_0, b_0), \end{aligned}$$

where $f_{ij}(\boldsymbol{\lambda}) = f(t_i, v_j; \boldsymbol{\lambda})$. We set $\mu_1, \cdots, \mu_{24}$ equal to the values in Bondarenko et al. (2004) and $\tau_l^2 = (\frac{1}{2}\log 3)^2$ for $l = 1, \cdots, 24$ (the choice for τ_l is based on our assumption that λ_l will be within $\mu_l \pm 3$ with 95% confidence). In the prior distribution of random error, we use $a_0 = 5$ and $b_0 = 12$.

The posterior distribution of the unknown parameters can be obtained from $p(\boldsymbol{\lambda}, \sigma_0^2|\boldsymbol{y}) \propto p(y|\boldsymbol{\lambda}, \sigma_0^2)p(\boldsymbol{\lambda})p(\sigma_0^2)$. We have

$$\begin{aligned} -\log p(\boldsymbol{\lambda}, \sigma_0^2|\boldsymbol{y}) &= \frac{mnK}{2}\log(\sigma_0^2) + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{K}\frac{(y_{ijk} - f_{ij}(\boldsymbol{\lambda}))^2}{2\sigma_0^2} \\ &\quad + \sum_{l=1}^{24}\left[\frac{1}{2}\log(\tau_l^2) + \frac{(\lambda_l - \mu_l)^2}{2\tau_l^2}\right] + (a_0 + 1)\log(\sigma_0^2) + \frac{b_0}{\sigma_0^2}, \end{aligned} \tag{25}$$

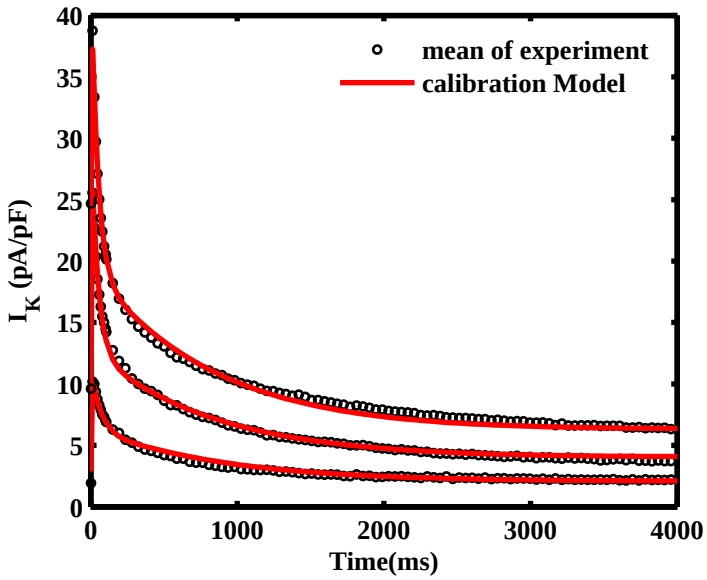

Figure 17: Mean of experiment from five apical cells vs. calibration model at v = -10, 20 and 50 mV.

where the normalizing constant is omitted. The posterior mode $(\hat{\boldsymbol{\lambda}}', \hat{\sigma}_0^2)'$ can obtained by minimizing $-\log p(\boldsymbol{\lambda}, \sigma_0^2|\boldsymbol{y})$. This gives us a plug-in version of the calibration model $f(t_i, v_j; \hat{\boldsymbol{\lambda}})$.

Figure 17 shows the mean of experimental measurements from the five cells and prediction from the calibration model. We can see that the calibration model agrees well with the mean of measurements. Therefore, the bias in the computer model is small and can be neglected. In Figure 18, we compare the calibration model and the Bondarenko model with the experimental measurements for each cell at 50 mV, 20 mV, and -10 mV. Clearly, calibration model gives improvement in prediction compared with the Bondarenko model. However, we can still observe discrepancy between calibration model and experimental data if we look at each cell separately. For example, the predictions are lower than the experimental measurements of cell 1 and cell 2, whereas higher than those of cells 3-5. This is because we used only one set of parameters in the calibration model for the five distinct cells. Even though the calibration model performs well on the average, the current amplitudes and decaying behavior of individual cells vary quite a lot. We can capture this variability by treating the calibration parameters as random effects as described in the next section.

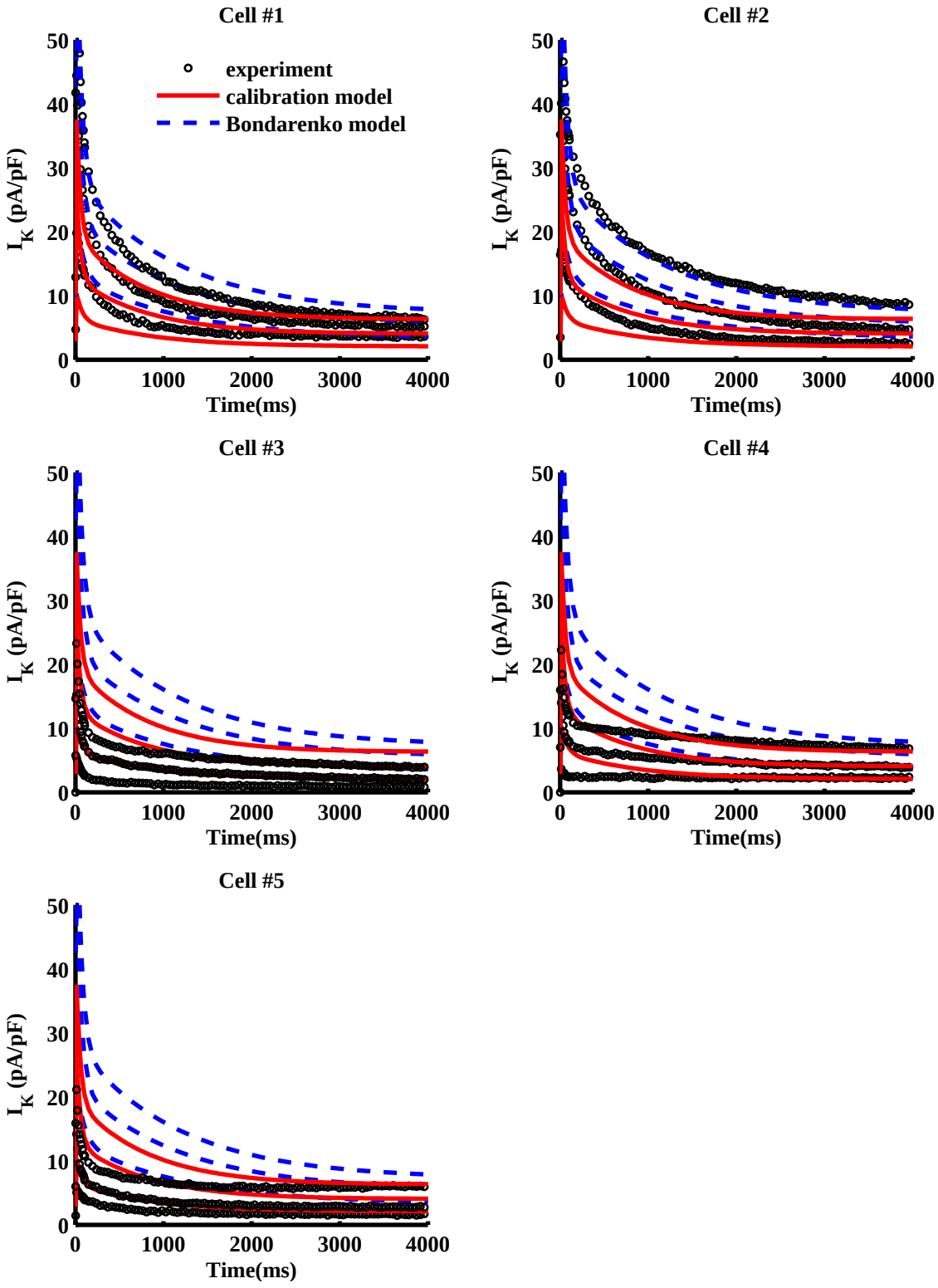

Figure 18: Experiment, calibration model and Bondarenko model for five apical cells at v = -10, 20 and 50 mV.

2.4.2 Calibration with Random Effects

There are a total of 24 calibration parameters in the model (24). Some of them could be affected by the cell dependent properties, such as cell size and resistance. To incorporate the cell-to-cell variability in the model, we can treat them as random effects. In a full random effects model with all the 24 calibration parameters treated as random effects, there are a total of 169 ($= 24 \times 5 + 24 + 24 + 1$) parameters.

A fully Bayesian approach of this high-dimensional nonlinear mixed effects model can be challenging (see, e.g., Lachos, Castro, and Dey 2013). However, in our problem, this full random effects model may be an unnecessarily large model in which several random effects may play the same role of explaining the cell-to-cell difference. Therefore, we propose a simple method to identify significant random effects and reduce the dimension of the problem. The idea of this method is similar to the forward variable selection algorithm in regression analysis. That is, first we find a parameter which if varied over the cells will give the maximum reduction in the mean squared error:

$$MSE_l = \min_{\boldsymbol{\lambda}_l} \frac{1}{mnK} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{K} \left[y_{ijk} - f(t_i, v_j; \lambda_{lk}, \hat{\boldsymbol{\lambda}}_{(-l)}) \right]^2,$$

where $\boldsymbol{\lambda}_l = (\lambda_{l1}, \cdots, \lambda_{lK})'$ are the lth parameter values for the K cells and $\hat{\boldsymbol{\lambda}}_{(-l)}$ the remaining set of parameters fixed at the estimated value. Computation of MSE_l requires K one-dimensional optimizations. MSE_l are plotted in Figure 19 for $l = 1, \ldots, 24$. We can see that λ_2 is the most important parameter for reducing the mean squared error. This procedure can be continued to identify the important parameters one-by-one.

Interestingly, λ_2 is the conductance of the second potassium channel, which is known to be influenced by the cell volume. Thus, it makes physical sense to use the same random effect for the other two conductances λ_1 and λ_3. Therefore, in the first step of the procedure, we chose to introduce a common random effect for all the three conductances $(\lambda_1, \lambda_2, \lambda_3)$. This gave a reduction of MSE from 12.58 to 1.33. Continuing the procedure, we identified λ_{20}, λ_{11}, λ_{14}, λ_{23}, etc. The corresponding MSE values are shown in Figure 19. We can see that the reduction in MSE after

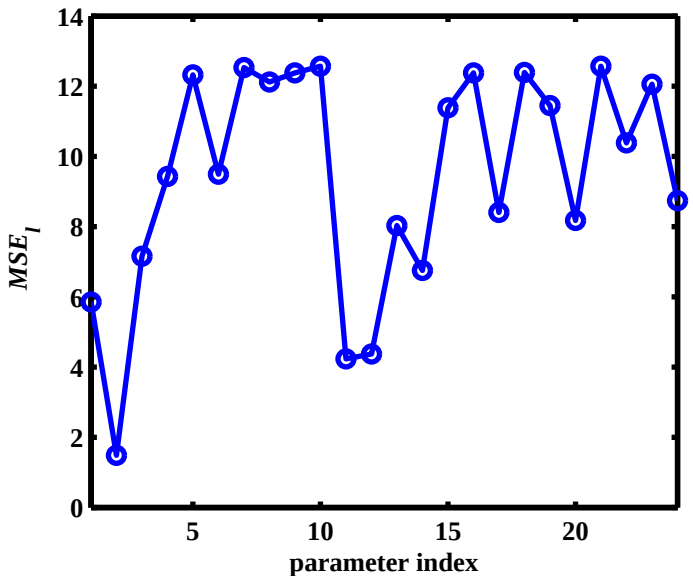

Figure 19: Plot of MSE_l against λ indices

step 2 is negligibly small. Therefore, we decided to include only the first two random effects in our model.

Thus, the proposed random effects model for cell k is

$$y_{ijk} = f(t_i, v_j;\, \beta_{1k} + \lambda_1, \beta_{1k} + \lambda_2, \beta_{1k} + \lambda_3, \lambda_4, \cdots, \lambda_{19}, \beta_{2k} + \lambda_{20}, \lambda_{21}, \cdots, \lambda_{24}) + \epsilon_{ijk}, \tag{26}$$

where β_{1k} is the common random effect for the three conductances (λ_1, λ_2, and λ_3) and β_{2k} is the random effect for λ_{20}. Assume $\boldsymbol{\beta}_k \sim^{iid} N(\mathbf{0}, \boldsymbol{\Sigma}_\beta)$, for $k = 1, \cdots, K$, where $\mathbf{0} = (0,0)'$ and $\boldsymbol{\Sigma}_\beta = diag(\sigma_1^2, \sigma_2^2)$. Using noninformative prior for σ_1^2 and σ_2^2, we have the following Bayesian model:

$$\begin{aligned}
y_{ijk}|\boldsymbol{\lambda}, \boldsymbol{\beta}_k, \sigma_0^2 &\sim^{ind} N(f(t_i, v_j; \boldsymbol{\lambda}, \boldsymbol{\beta}_k), \sigma_0^2), \\
\lambda_l &\sim^{iid} N(\mu_l, \tau_l^2), l = 1, \cdots, 24 \\
\boldsymbol{\beta}_k &\sim^{iid} N(\mathbf{0}, \boldsymbol{\Sigma}_\beta), \\
\sigma_1^2, \sigma_2^2 &\sim^{iid} IG(0.001, 0.001), \\
\sigma_0^2 &\sim IG(a_0, b_0).
\end{aligned} \tag{27}$$

This model contains $d = 37 (= 2 \times 5 + 24 + 2 + 1)$ unknown parameters. This is still a high dimensional model, but much smaller than the initial 169-dimensional model.

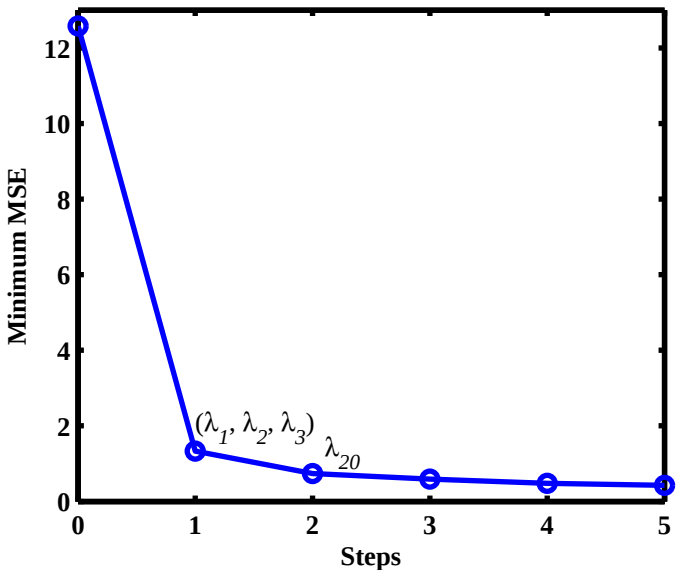

Figure 20: MSE in the first five steps

We can use MCMC methods to sample from the posterior and compute the desired posterior quantities. Let $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \cdots, \boldsymbol{\beta}_K')'$ denote all the random effects in the model. The posterior samples $(\boldsymbol{\lambda}^{(l)}, \boldsymbol{\beta}^{(l)}, \sigma_1^{2(l)}, \sigma_2^{2(l)}, \sigma_0^{2(l)})$, $l = 1, \cdots, N$ are obtained using Metropolis-within-Gibbs algorithm (Gelman, et al. 2004). The sample size is chosen to be $N = 20000 \times d$. The details of the MCMC algorithm are given in the Appendix.

The mean prediction (for the population of cells) at t_i and v_j can be obtained as

$$\frac{1}{NK} \sum_{l=1}^{N} \sum_{k=1}^{K} f(t_i, v_j; \boldsymbol{\lambda}^{(l)}, \boldsymbol{\beta}_k^{(l)}).$$

The credible interval of the mean prediction can be obtained by computing appropriate quantiles of $f(t_i, v_j; \boldsymbol{\lambda}^{(l)}, \boldsymbol{\beta}_k^{(l)})$ at each t_i and v_j. Figure 21 shows the mean prediction of the random effects model and 95% credible intervals using the MCMC sample at $v = 50$ mV . Compared with Bondarenko model, our random effects model more accurately predicts the overall trend of the current.

Similarly, the mean prediction at t_i and v_j for cell k is given by

$$\frac{1}{N} \sum_{l=1}^{N} f(t_i, v_j; \boldsymbol{\lambda}^{(l)}, \boldsymbol{\beta}_k^{(l)}).$$

The credible intervals can be obtained as before. The mean prediction for each cell

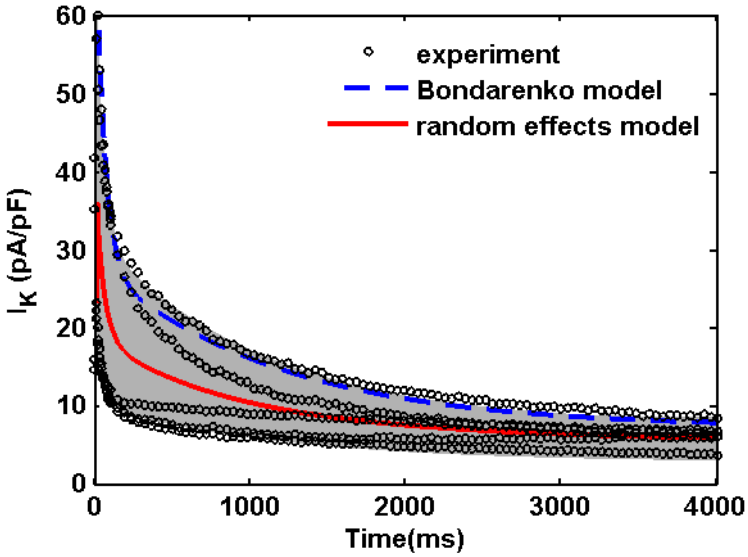

Figure 21: Predictions of random effects model and 95% credible interval at $v = 50$ mV

is compared with Bondarenko model and experimental measurements at $v = -10, 20$ and 50 mV in Figure 22. We can see that the random effects model greatly reduces the discrepancy between predictions and experimental measurements for each individual cell.

2.5 Conclusions

This chapter discussed the calibration of a complex computer model for predicting the potassium currents in a cardiac cell. Some nonstandard techniques were employed to tackle this challenging problem. In the existing literature, a complex computer model is treated as a black box and a statistical model such as kriging is fitted based on the data from a carefully designed computer experiment. We couldn't succeed using this standard approach. But we were able to approximate the computer model by making some simple approximations inside the model. This required careful study and understanding of the underlying mathematical model, but it was worth it. It exemplifies the benefits of integrating mathematical/statistical modeling with physical/engineering knowledge.

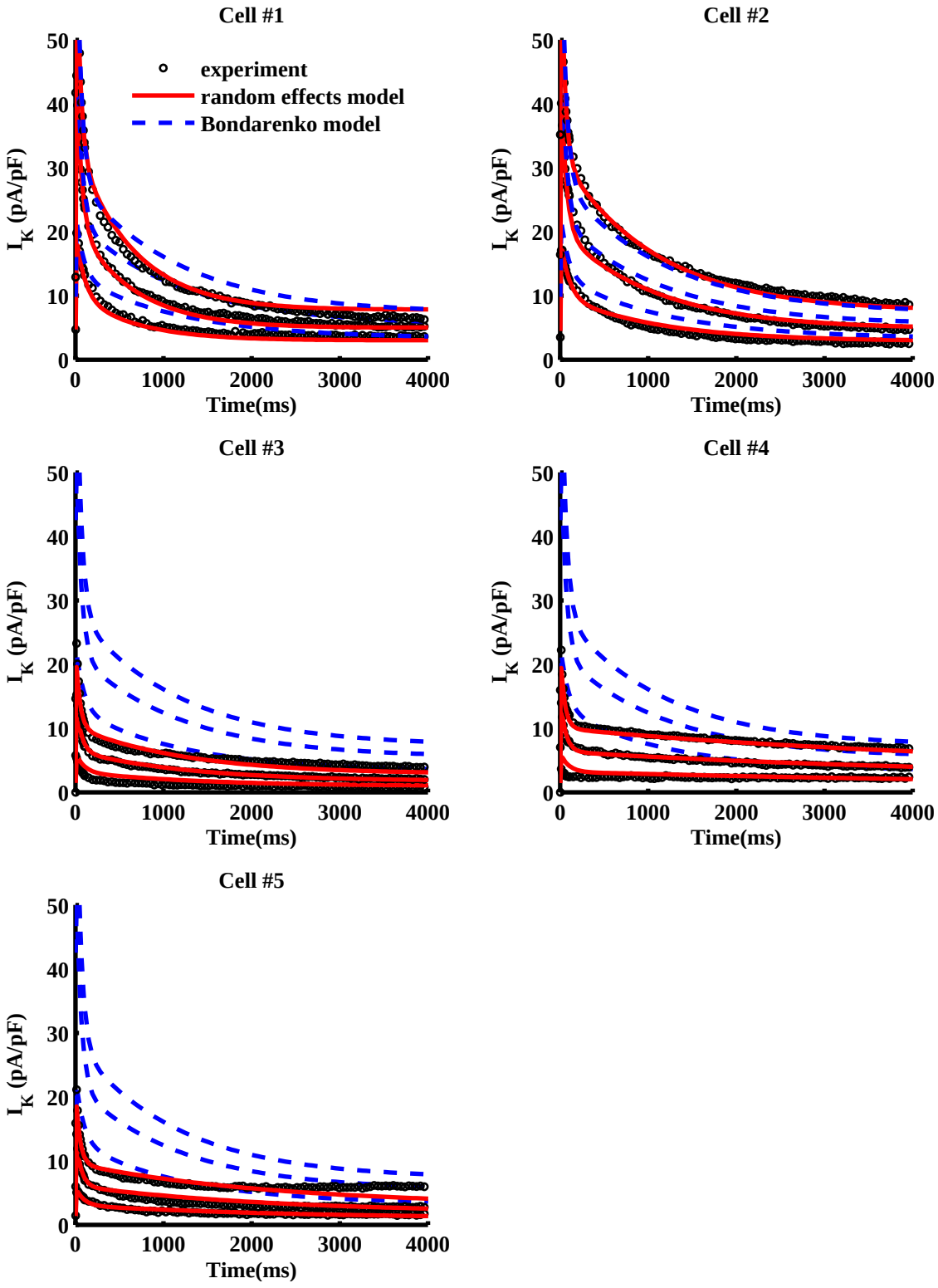

Figure 22: Experiment, random effects model and Bondarenko model for five apical cells at v = -10, 20 and 50 mV

We also developed a simple and intuitive approach for identifying important random effect parameters in a nonlinear mixed effects model by borrowing ideas from regression analysis. It greatly reduced the dimensionality of the problem enabling us to solve the problem quite efficiently. More than calibrating a particular cardiac cell model to a given set of data, we were able to develop a methodology that can be used in solving similar problems in cardiac modeling and possibly in other areas.

2.6 Acknowledgements

The authors would like to acknowledge the support to this present work from the U.S. National Science Foundation grants (CMMI-1266025 & 1266331), and also thank Dr. Andrew R. Ednie and Eric S. Bennett for sharing the datasets from electrophysiological experiments.

2.7 Appendix

2.7.1 K^+ Dynamics in Markov model (Bondarenko et al. 2004)

$$\frac{d[K^+]_i}{dt} = -(I_{Kto,f} + I_{Kss} + I_{Kur})\frac{A_{cap}C_m}{V_{myo}F} \tag{A-1}$$

Transient outward K^+ current $I_{Kto,f}$

$$\begin{aligned}
I_{Kto,f} &= \eta_1 a_{to,f}^3 i_{to,f}(v - E_K) && \text{(A-2)}\\
E_K &= \frac{RT}{F}\ln(\frac{[K^+]_o}{[K^+]_i}) && \text{(A-3)}\\
\frac{da_{to,f}}{dt} &= \alpha_a(1 - a_{to,f}) - \beta_a a_{to,f} && \text{(A-4)}\\
\frac{di_{to,f}}{dt} &= \alpha_i(1 - i_{to,f}) - \beta_i i_{to,f} && \text{(A-5)}\\
\alpha_a &= \eta_4 e^{\eta_5 v} && \text{(A-6)}\\
\beta_a &= \eta_6 e^{-\eta_7 v} && \text{(A-7)}\\
\alpha_i &= \frac{\eta_8 e^{-v/\eta_9}}{\eta_{10}e^{-v/\eta_9} + 1} && \text{(A-8)}\\
\beta_i &= \frac{\eta_{11} e^{v/\eta_9}}{\eta_{12}e^{v/\eta_9} + 1} && \text{(A-9)}
\end{aligned}$$

Ultrarapidly activating delayed rectifier K^+ current

$$
\begin{aligned}
I_{Kur} &= \eta_2 a_{ur} i_{ur} (v - E_K) && \text{(A-10)} \\
\frac{da_{ur}}{dt} &= \frac{a_{ss} - a_{ur}}{\tau_{aur}} && \text{(A-11)} \\
\frac{di_{ur}}{dt} &= \frac{i_{ss} - i_{ur}}{\tau_{iur}} && \text{(A-12)} \\
a_{ss} &= 1/[1 + e^{-(v+\eta_{13})/\eta_{14}}] && \text{(A-13)} \\
\tau_{aur} &= \eta_{15} e^{-\eta_{16} v} + \eta_{17} && \text{(A-14)} \\
i_{ss} &= 1/[1 + e^{-(v+\eta_{18}))/\eta_{19}}] && \text{(A-15)} \\
\tau_{iur} &= \eta_{20} - \frac{\eta_{21}}{1 + e^{(v+\eta_{18})/\eta_{19}}} && \text{(A-16)}
\end{aligned}
$$

Noninactivating steady-state K^+ current

$$
\begin{aligned}
I_{Kss} &= \eta_3 a_{Kss} i_{Kss} (v - E_K) && \text{(A-17)} \\
\frac{da_{Kss}}{dt} &= \frac{a_{ss} - a_{Kss}}{\tau_{Kss}} && \text{(A-18)} \\
\frac{di_{Kss}}{dt} &= 0 && \text{(A-19)} \\
\tau_{Kss} &= \eta_{22} e^{-\eta_{23} v} + \eta_{24} && \text{(A-20)}
\end{aligned}
$$

2.7.2 Initial values of gating variables

Table 2: Initial values of gating variables

Parameter	Value
$a_{to,f}$	0.265563×10^{-2}
$i_{to,f}$	0.999977
a_{ur}	0.417069×10^{-2}
i_{ur}	0.998543
a_{Kss}	0.417069×10^{-3}
i_{Kss}	1.0

2.7.3 MCMC Method for Posterior Inference

Let $\boldsymbol{y}$ be the vector of all the nmK observations in the experiment and $\boldsymbol{y}_k$ be the vector of nm observations for the kth cell. The full conditional distributions for the

unknown parameters $(\boldsymbol{\lambda}', \sigma_1^2, \sigma_2^2, \sigma_0^2, \boldsymbol{\beta})$ are given by

$$
\begin{aligned}
p(\boldsymbol{\lambda}|\boldsymbol{y}, \sigma_1^2, \sigma_2^2, \sigma_0^2, \boldsymbol{\beta}) &\propto p(\boldsymbol{y}|\boldsymbol{\lambda}, \sigma_1^2, \sigma_2^2, \sigma_0^2, \boldsymbol{\beta}) \cdot p(\boldsymbol{\lambda}), \\
&\propto \exp\left\{-\frac{1}{2\sigma_0^2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{K}\left[y_{ijk} - f(t_i, v_j; \boldsymbol{\lambda}, \boldsymbol{\beta}_k)\right]^2\right\} \cdot \\
&\quad \exp\left[-\sum_{l=1}^{24}\frac{(\lambda_l - \mu_l)^2}{2\tau_l^2}\right], \\
p(\sigma_0^2|\boldsymbol{y}, \boldsymbol{\lambda}, \sigma_1^2, \sigma_2^2, \boldsymbol{\beta}) &\sim IG(a_0 + \frac{mnK}{2}, b_0 + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{K}\left[y_{ijk} - f(t_i, v_j; \boldsymbol{\lambda}, \boldsymbol{\beta}_k)\right]^2), \\
p(\sigma_1^2|\boldsymbol{y}, \boldsymbol{\lambda}, \sigma_0^2, \sigma_2^2, \boldsymbol{\beta}) &\sim IG(a_1 + \frac{K}{2}, b_1 + \frac{1}{2}\sum_{k=1}^{K}\beta_{1k}^2), \\
p(\sigma_2^2|\boldsymbol{y}, \boldsymbol{\lambda}, \sigma_0^2, \sigma_1^2, \boldsymbol{\beta}) &\sim IG(a_2 + \frac{K}{2}, b_2 + \frac{1}{2}\sum_{k=1}^{K}\beta_{2k}^2), \\
p(\boldsymbol{\beta}_k|\boldsymbol{y}, \boldsymbol{\lambda}, \sigma_0^2, \sigma_1^2, \sigma_2^2) &\propto p(\boldsymbol{y}_k|\boldsymbol{\lambda}, \sigma_1^2, \sigma_2^2, \sigma_0^2, \boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}_k|\sigma_1^2, \sigma_2^2) \\
&\propto \exp\left\{-\frac{1}{2\sigma_0^2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[y_{ijk} - f(t_i, v_j; \boldsymbol{\lambda}, \boldsymbol{\beta}_k)\right]^2\right\} \cdot \\
&\quad \exp\left[-(\frac{\beta_{1k}^2}{2\sigma_1^2} + \frac{\beta_{2k}^2}{2\sigma_2^2})\right], k = 1, \cdots, K.
\end{aligned}
$$

2.7.4 Calibration Results

Table 3: Calibrated parameters for ST3Gal4-\- mouse apical cell (* the difference between the calibration estimates from random effects model and the values in Bondarenko et al. 2004; ** the mean of the five cells is presented)

Variable	Parameter	Bondarenko et al. 2004	Initial calibration estimates	Random effects model	Difference* (%)
Conductance	η_1	0.4067	0.2145	0.1844**	-55%
	η_2	0.1156	0.1079	0.1307**	-18%
	η_3	0.050	0.0491	0.0365**	-27%
$a_{to,f}$	η_4	0.5283	0.2589	0.2581	-51%
	η_5	0.03577	0.0221	0.0208	-42%
	η_6	0.0609	0.0325	0.0247	-59%
	η_7	0.06237	0.0728	0.0780	+25%
$i_{to,f}$	η_8	2.21×10^{-5}	2.20×10^{-5}	1.54×10^{-4}	+598%
	η_9	7.00	8.19	20.29	+190%
	η_{10}	5.60×10^{-4}	5.60×10^{-4}	4.85×10^{-4}	-13%
	η_{11}	0.1138	0.1032	0.0246	-78%
	η_{12}	6.15	4.38	1.02	-83%
a_{ur}	η_{13}	22.50	14.10	5.349	-76%
	η_{14}	7.7	33.89	34.03	+342%
	η_{15}	0.493	0.534	1.005	+103%
	η_{16}	0.0629	0.060	0.048	-23%
	η_{17}	2.058	3.023	2.689	31%
i_{ur}	η_{18}	45.20	38.25	40.59	-10%
	η_{19}	5.7	5.4	9.97	+75%
	η_{20}	1200	858	1316**	+10%
	η_{21}	170	174	763	+349%
a_{Kss}	η_{22}	39.3	38.5	98.4	+150%
	η_{23}	0.0862	0.1055	0.0646	-25%
	η_{24}	13.17	17.68	2.70	-80%

CHAPTER III

UNCERTAINTY QUANTIFICATION IN SOLID END MILLING PROCESS

3.1 Introduction

Computer models are developed to simulate and predict the behavior of the systems which are too complex for analytical solutions. The computer model itself is deterministic, but if the inputs have uncertainties, variability could be observed on the outputs. The study of uncertainty propagation in computer model aims at finding out how the uncertainties of inputs will be transmitted to the model outputs. Understanding uncertainty propagation is critical in identifying the confidence of an outcome and therefore the key for addressing decision-making problems.

There are basically two categories of methodologies for studying uncertainty propagation: intrusive methods and non-intrusive methods. Common intrusive methods include perturbation method (Tang and Pinder, 1977, Detfinger and Wilson, 1981), Neumann series expansion (Zeitoun and Braester, 1991), stochastic Galerkin (Ghanem and Spanos, 1991), etc. The basic idea of intrusive methods is to incorporate stochastic expansion for the uncertainty variable and transform the stochastic problem into a deterministic problem. Since the intrusive methods have a close connection to original computer model, it has a strong physical interpretation and customization. However, in practice intrusive methods are challenging and expensive to implement for many complex models, because extensive modifications are required in the original computer codes and usually a much larger group of coupled and deterministic equations with respect to the original computer model need to be solved.

Non-intrusive methods treat the computer model as a black box and only require a collection of outputs from the model. Monte Carlo (MC) method may be the most intuitive and widely used non-intrusive method for uncertainty analysis. The basic

idea of MC method is to create a mapping between inputs and outputs based on a random sampling of inputs and obtain the probability distribution of the system output empirically. MC method is simple, but it requires extensive sampling of inputs for convergence, especially in high dimensional cases. It would be very computationally expensive if the evaluation of computer model is time consuming. To improve the computational efficiency, other sampling techniques are developed, such as Quasi-Monte Carlo (QMC) (Fang et al. 2006). In QMC, low discrepancy sequences are used for sampling. Since low discrepancy sequences have better uniformity than pseudo-random sequence, the convergence of QMC is faster than that of MC as the number of runs increases. A thorough review of sampling approaches is given by Helton and Davis (2003).

If the computer model is expensive to evaluate, the aforementioned methods could be time consuming. One idea is to build an easy-to-evaluate metamodel to approximate and surrogate the complex computer model in uncertainty analysis. Differential analysis method (Tomovic and Vukobratovic 1972, Lewins and Becker 1982, Rabitz et al. 1983, Ronen 1988, Turányi 1990) and response surface methodology (Aceil and Edwards 1991, Lee et al. 1987, and Kim et al. 1986) are two early developed approximation methods. The idea of differential analysis method is to use the Taylor series expansion near a base-case set of input. Response surface methodology is based on using an experimental design to construct an appropriate response surface approximation. But both approaches could not be sufficiently accurate if the system has strong nonlinearity or the input uncertainty is large. With the development in design and analysis of computer experiments (Santner et al. 2003), metamodels using kriging (Sacks, et al., 1989) are able to provide a better approximation for the complex computer models. They can be used to surrogate the computer model in MC or QMC analysis (Chen et al. 1997 and Delaurentis and Mavris, 2000). Obviously, the accuracy of uncertainty quantification strongly depends on the prediction performance of metamodel. However, building a good metamodel for a complex system is still a challenging problem.

This chapter studies the uncertainty propagation in the computer model used for

simulating the solid end milling process in the software Production Module. Production Module is developed by the company Third Wave Systems and widely used for the simulation and optimization in various types of machining processes. In this work, we focus on solid end milling process. The uncertainties come from the material properties of workpiece and tool geometries, and propagate to the output, cutting force. MC and QMC approaches are initially attempted to study this uncertainty propagation problem. But we found MC method to be too slow and QMC to be time consuming. Then we developed an ordinary kriging model for solid end milling process. Unfortunately, large discrepancies were observed between the predictions from the kriging model and the true outputs. Therefore, we have to look for an alternative approach to solve the problem.

A local base emulator was proposed to model the cutting force and study the uncertainty propagation. The basic idea is to use a base function, which is the simulation output with the uncertainty parameters set at their nominal values, and fit an appropriate model to capture the discrepancy between the base and true outputs. Linear function and Gaussian process (GP) are used for modeling the discrepancy and both are found to work well.

This chapter is organized as follows. In Section 3.2, we briefly introduce the computer model for solid end milling process and specify the uncertainty sources. In Section 3.3, we review three traditional methodologies of uncertainty analysis. In Section 3.4, we describe the methodologies of linear base emulator and Gaussian process base emulator. Some concluding remarks and discussions are given in Section 3.5.

3.2 Computer Model of Solid End Milling Process

The solid end mill is a widely used machining tool type in machining process. Production Module is a software based on physics-based material models, which provides a computer model to simulate the solid end milling process. In the computer model, input variables involve feed rate, axial depth of cut, and radial depth of cut. They will change as functions of time in a simulation. The workpiece properties and

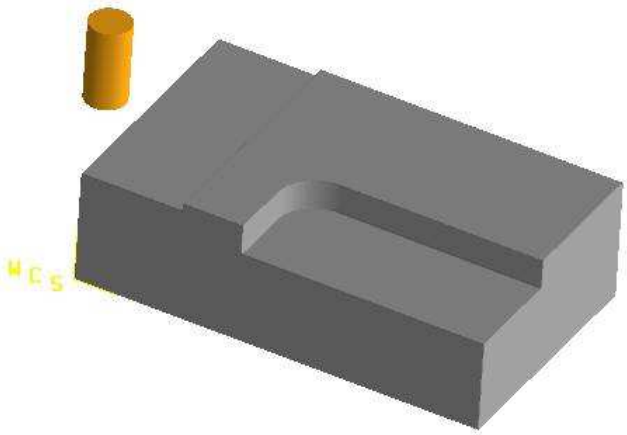

Figure 23: Tool path of solid end milling example

tool geometries can be specified by user defined parameters, including the hardness of the workpiece material, the rake angle, helix angle, relief angle, and corner radius of the tool. In practice, the properties of the workpiece and tool geometry may have certain uncertainties. Understanding the uncertainty propagation is critical for optimizing the tool path and other machining settings. In Production Module, multiple outputs can be generated from a simulation. In this chapter, we will only consider the peak tangential force, which is a key force in the machining process. The uncertainty quantification of the other outputs can be dealt in a similar way.

Throughout this chapter, the demo of solid end milling process in Figure 23 will be used as an example to illustrate and compare different methodologies. Figure 24 shows the simulated peak tangential force from Production Module with all the parameters at their nominal values. There are a total of 2885 time points in this simulation. At each time point, the simulation can be seen as a short straight cutting.

3.3 Review of Methodologies for Uncertainty Analysis

In this section, we will go over the three popular methods for uncertainty propagation: MC method, QMC method, and metamodel approach. To begin with, we first define the notations which will be used throughout the work. Consider a computer model

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

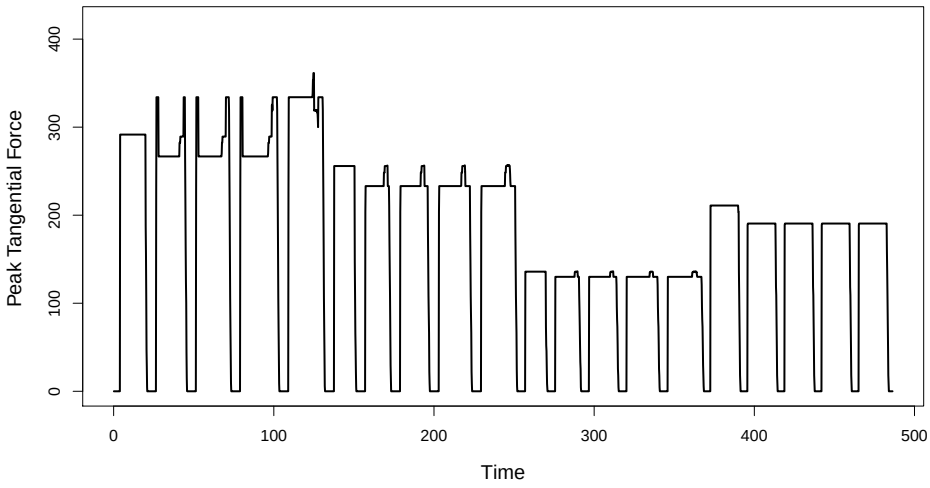

Figure 24: Peak tangential force predicted by Production Module for the solid end milling process in Figure 23

where y denotes the output of the computer model, $\boldsymbol{x} = (x_1, \cdots, x_p)'$ is the vector of input variables with no uncertainties, and $\boldsymbol{u} = (u_1, \cdots, u_q)'$ represents the model parameters with uncertainties. In the solid end milling demo as introduced in Section 2, we have three input variables with no uncertainties ($p = 3$) and five parameters with uncertainties ($q = 5$). The three input variables are feed rate (x_1), axial depth of cut (x_2), and radial depth of cut (x_3), and the five parameters are the hardness of the workpiece material (u_1), rake angle (u_2), helix angle (u_3), relief angle (u_4), and corner radius (u_5). The nominal values of the five parameters are 111 Bhn, 10 deg, 20 deg, 10 deg, and 0 inch, respectively. The uncertainties of $u_1, \cdots, u_q$ can be characterized by a distribution, $p(\boldsymbol{u})$ which is known. Since $\boldsymbol{u}$ has uncertainties, $\boldsymbol{u}$ could deviate from its nominal value $\boldsymbol{u}_0$. Our objective is to study how the uncertainty will be transmitted from $\boldsymbol{u}$ to the output y.

In the solid end milling example, let the vector $\boldsymbol{u}_0 = (111, 10, 20, 10, 0)'$ denote the nominal value of $\boldsymbol{u}$. The parameters $u_1, \cdots, u_5$ are assumed to be independently distributed, which is a reasonable assumption in practice. We also assume $u_1, \cdots, u_4$ follow normal distributions with means equal to their nominal values and standard deviations equal to 15% of their nominal values. Corner radius, u_5 is assumed to

follow an exponential distribution with mean equals to 0.01.

3.3.1 Monte Carlo method

In MC analysis, we first need to generate a sample of $\boldsymbol{u}$ by using the sequences of pseudo-random numbers. Suppose we have generated a sample, $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N$. Then, the expected value and variance are approximately

$$\begin{aligned} E(y) &\approx \bar{f}(\boldsymbol{x}) = \frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{x}, \boldsymbol{u}_j), \\ Var(y) &\approx \frac{1}{N-1}\sum_{j=1}^{N}[f(\boldsymbol{x}, \boldsymbol{u}_j) - \bar{f}(\boldsymbol{x})]^2. \end{aligned} \tag{28}$$

The confidence interval of y can be obtained by computing appropriate quantiles of $f(\boldsymbol{x}, \boldsymbol{u}_j)$, $j = 1, \cdots, N$.

For the solid end milling example, we need a large sample (run size > 1000) to obtain convergence. Each evaluation of the solid end milling demo costs $10 \sim 15$ seconds, which means the computation will take several hours. Since MC method is too time consuming, we will not implement it.

3.3.2 Quasi–Monte Carlo method

The difference between MC and QMC is the sampling technique. As mentioned ealier, QMC method is based on low-discrepancy sequences instead of pseudo-random numbers. For the solid end milling example, we first generated a uniform design with sample size $50(= 10 \times q)$ in $[0, 1]^5$. The design of $\boldsymbol{u}$, $\boldsymbol{D_u} = \{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{50}\}$ can be obtained by using the inverse probability transformation.

Then we run the solid end milling simulation with $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{50}$, respectively. The 50 peak tangential force profiles are shown in Figure 25. Computing appropriate quantiles using the profiles in Figure 25, we can obtain the confidence interval of the simulation. The 95 % confidence interval of peak tangential force in the simulation is shown in Figure 26. Simulating 50 times takes about 15 minutes, which is much faster than the MC method. For a simulation with a more complex tool path, say curve cutting, the run-time could be larger. Then QMC method can become very time

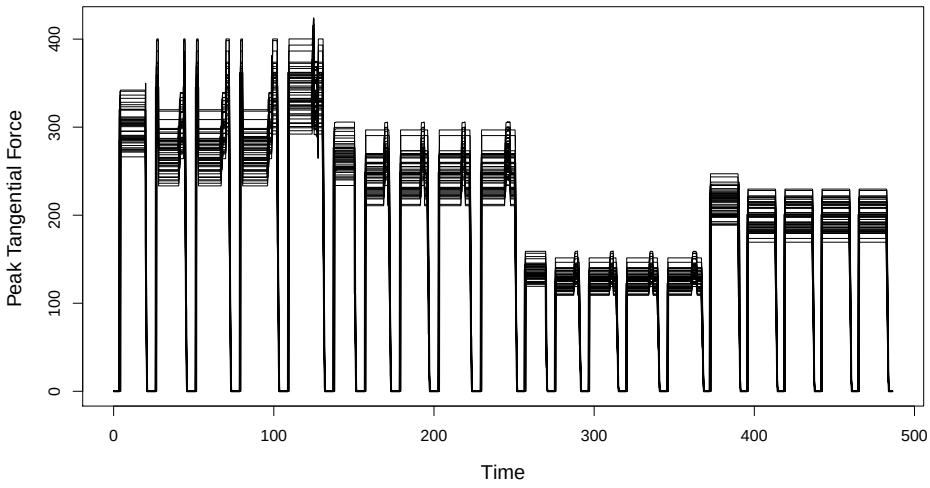

Figure 25: 50 simulated peak tangential force profiles for the QMC.

consuming. Therefore, it's necessary to find another way to improve the efficiency of uncertainty analysis.

3.3.3 Metamodel Method

The efficiency of MC or QMC method is limited by the the computational expense of computer model. Using an easy-to-evaluate metamodel to surrogate the computer model can reduce the total computation time. Ordinary kriging (Fang et al. 2006) is a widely used metamodeling approach for the approximation of complex computer model.

Suppose we have an N-run experimental design $\boldsymbol{D}_{\boldsymbol{x},\boldsymbol{u}} = ((\boldsymbol{x}_1', \boldsymbol{u}_1'), \cdots, (\boldsymbol{x}_N', \boldsymbol{u}_N'))'$ and the $N \times 1$ vector $\boldsymbol{y} = (y_1, \cdots, y_N)'$ is the collection of the outputs by conducting the experiment. The ordinary kriging model is given by

$$\begin{aligned} y(\boldsymbol{x}, \boldsymbol{u}) &= \mu + Z(\boldsymbol{x}, \boldsymbol{u}), \\ Z(\boldsymbol{x}, \boldsymbol{u}) &\sim GP(0, \sigma^2 R(\cdot, \cdot)), \end{aligned} \tag{29}$$

where σ^2 denotes the variance, and the correlation function is defined as

$$corr(Z(\boldsymbol{x}_i, \boldsymbol{u}_i), Z(\boldsymbol{x}_j, \boldsymbol{u}_j)) = R(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{u}_i - \boldsymbol{u}_j) = R_{\boldsymbol{x}}(\boldsymbol{x}_i - \boldsymbol{x}_j) R_{\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_j).$$

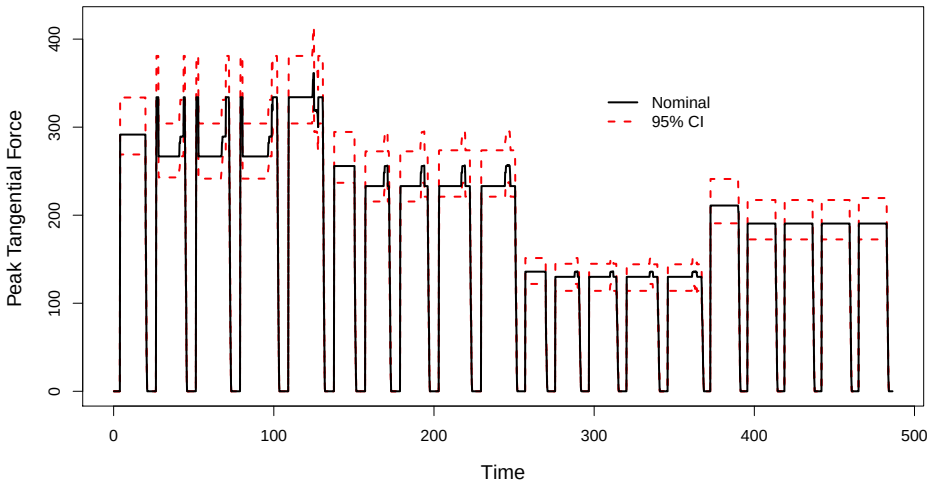

Figure 26: The simulation at nominal value of parameters and the 95% confidence interval obtained from QMC

The Gaussian correlation functions are used for $R_{\boldsymbol{x}}(\cdot)$ and $R_{\boldsymbol{u}}(\cdot)$:

$$R_{\boldsymbol{x}}(\boldsymbol{x}_i - \boldsymbol{x}_j) = \exp\{-\sum_{k=1}^{p} \theta_k (x_{ik} - x_{jk})^2\},$$

$$R_{\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_j) = \exp\{-\sum_{k=1}^{q} \phi_k (u_{ik} - u_{jk})^2\}.$$

The predictor is given by

$$\hat{y}(\boldsymbol{x}, \boldsymbol{u}) = \hat{\mu} + \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{u})' \boldsymbol{R}^{-1} (\boldsymbol{y} - \hat{\mu}\mathbf{1}), \tag{30}$$

where $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{u}) = (R(\boldsymbol{x} - \boldsymbol{x}_1, \boldsymbol{u} - \boldsymbol{u}_1), \cdots, R(\boldsymbol{x} - \boldsymbol{x}_N, \boldsymbol{u} - \boldsymbol{u}_N))'$, $\boldsymbol{R}$ is an $N \times N$ matrix with ijth element $R(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{u}_i - \boldsymbol{u}_j)$, and $\hat{\mu} = \mathbf{1}'\boldsymbol{R}^{-1}\boldsymbol{y}/\mathbf{1}'\boldsymbol{R}\mathbf{1}$. Let $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)'$ and $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_q)'$. The correlation parameters $(\boldsymbol{\theta}', \boldsymbol{\phi}')'$ can be estimated by minimizing the negative log-likelihood function

$$N \log \hat{\sigma}^2 + \log |\boldsymbol{R}|, \tag{31}$$

where $\hat{\sigma}^2 = \frac{1}{N}(\boldsymbol{y} - \hat{\mu}\mathbf{1})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \hat{\mu}\mathbf{1})$. After we obtain the fitted ordinary kriging model, we can use it to replace the computer model in MC or QMC method for uncertainty analysis.

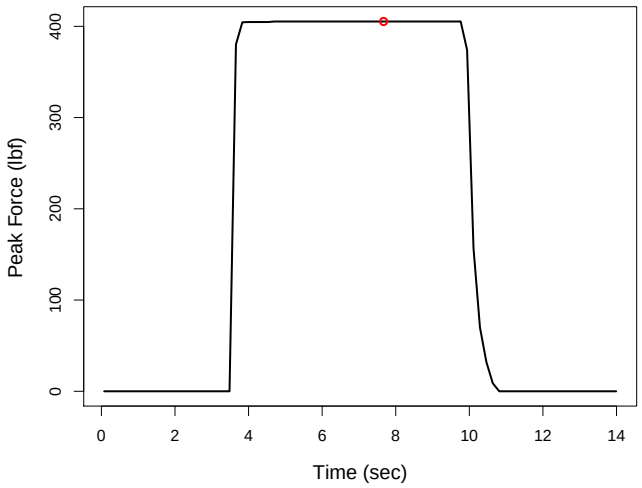

Figure 27: Simulation of a simple straight cutting

To fit an ordinary kriging model for the solid end milling process, we first need to generate a design of experiment by varying $\boldsymbol{x}$ and $\boldsymbol{u}$. Notice that, here we do not need to run the whole simulation for the experimental design, which will make the generation of design time consuming. We only need to run a simple straight cutting simulation to obtain the output for each run in the design. For example, the top straight line between between 4 seconds and 10 seconds in Figure 27 is the simulation of a straight cutting at a fixed set of $(\boldsymbol{x}, \boldsymbol{u})$. Then the corresponding tangential cutting force can be obtained from any point on the straight line (e.g. the red circle in Figure 27). Since the simulation of a short straight cutting is very fast, the computational expense of performing experimental design is small.

We use a cross array for the design of straight cutting experiment. Let $\boldsymbol{D_x} = (\boldsymbol{x}_1', \cdots, \boldsymbol{x}_m')'$ and $\boldsymbol{D_u} = (\boldsymbol{u}_1', \cdots, \boldsymbol{u}_n')'$ denote the designs for $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively. The whole design $D_{\boldsymbol{x},\boldsymbol{u}}$ is obtained as $\boldsymbol{D_{x,u}} = \boldsymbol{D_x} \otimes \boldsymbol{D_u}$, where $\otimes$ denotes the Kronecker product. For $\boldsymbol{D_u}$, we can first generate an n-dimensional Maximum Projection Latin hypercube design (Joseph, et. al. 2014) in $[0, 1]$ and then transform the design to the original scale by using the probability distribution of $\boldsymbol{u}$.

For generating $\boldsymbol{D_x}$, we need to be more careful. Let $\boldsymbol{\mathcal{X}}_0 = \{\boldsymbol{x} : f(\boldsymbol{x}, \boldsymbol{u}_0)\}$ be the set of $\boldsymbol{x}$ in the simulation. Notice that only the outputs with $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_0$ are related to

uncertainty analysis. Therefore, instead of generating $\boldsymbol{D_x}$ from the global space of $\boldsymbol{x}$, we generate it within the local space $\boldsymbol{\mathcal{X}}_0$. Our idea is to modify an appropriate existing space-filling design criterion to construct $\boldsymbol{D_x}$. Maximum Projection (MaxPro) design (Joseph, et. al. 2014) maximizes space-filling properties on the projections to all the subsets of factors and was found to have certain advantages for using in computer experiment over the other space-filling designs. Below, we specify the steps of generating $\boldsymbol{D_x}$ by extending the criterion used in MaxPro design:

- Let $\boldsymbol{x}_1 = \underset{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}_0}{\arg\max} f(\boldsymbol{x}, \boldsymbol{u}_0)$ and use $\boldsymbol{x}_1$ as the first design point in $\boldsymbol{D_x}$
- Suppose $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{i-1}\} \subset \boldsymbol{D_x}$, $2 \leq i < m$ has been generated. Then, the ith point $\boldsymbol{x}_i$ can be found by using the criterion

$$\boldsymbol{x}_i = \arg \min_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}_0} \sum_{l=1}^{i-1} \frac{1}{\prod_{k=1}^{p}(x_k - x_{lk})^2}$$

- Repeat step 2 from $i = 2$ to $i = m$. The generated array $(\boldsymbol{x}_1', \cdots, \boldsymbol{x}_m')'$ is the m-run MaxPro design in $\boldsymbol{\mathcal{X}_0}$.

The next step is to fit the ordinary kriging model. Notice that, we have $\boldsymbol{R}^{-1} = \boldsymbol{R}_{\boldsymbol{x}}^{-1} \otimes \boldsymbol{R}_{\boldsymbol{u}}^{-1}$, where $\boldsymbol{R_x}$ is an m-by-m correlation matrix corresponding to $\boldsymbol{x}$ with ijth element $R_{\boldsymbol{x}}(\boldsymbol{x}_i - \boldsymbol{x}_j)$ and $\boldsymbol{R_u}$ is an n-by-n matrix with ijth element $R_{\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_j)$. Consequently, the computational complexity involving the inverse of $\boldsymbol{R}$ is reduced from $O(m^3n^3)$ to $O(n^3 + m^3)$ (Hung, Joseph, and Melkote 2014).

The coefficient parameters are obtained by minimizing the negative likelihood function (31). We obtain $\hat{\boldsymbol{\theta}} = (0.001, 0.880, 22.794)'$ and $\hat{\boldsymbol{\phi}} = (0.816, 0.106, 3.328, 0.831, 1.203)'$. In Figure 28, the prediction of the fitted kriging model is compared with the simulation from the computer model at the nominal setting. Unfortunately, we find the fitted metamodel gives a poor prediction. We can even observe that the ordinary kriging gives a nonzero prediction when the axial depth of cut or radial depth of cut equals to zero, which violates the basic physics in the machining process. Due to the poor accuracy of the kriging model, it will give an unreliable result in uncertainty analysis.

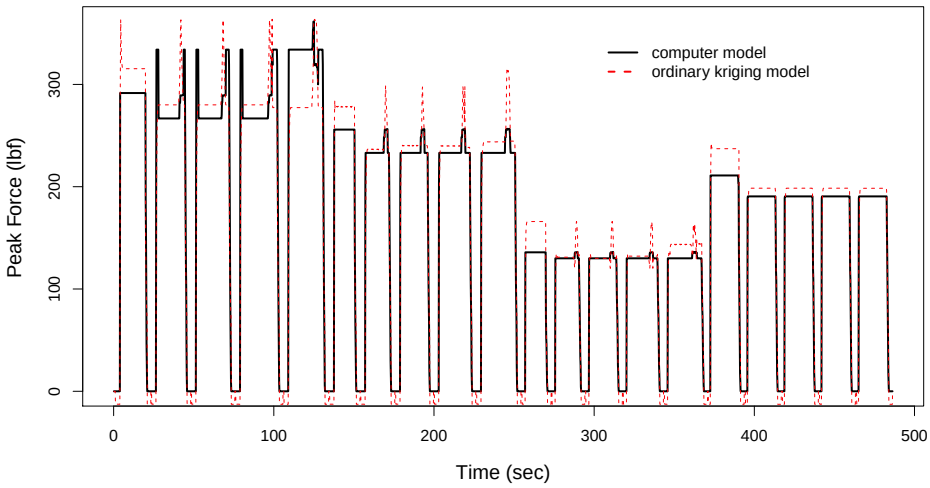

Figure 28: The predictions of ordinary kriging model at $\boldsymbol{u}_0$

3.4 Local Base Emulator

To improve the accuracy of the metamodel, one option is to use an experiment with much larger run size. But performing a larger design and evaluating kriging model could be time consuming. Therefore, we need to find an alternative way to improve the performance of metamodel.

We propose to make use of another information to improve the metamodeling. That is the outputs of simulation with the parameters equal to the nominal values $\boldsymbol{u}_0$. These outputs are known, because users will first run the computer model with the nominal parameter values and then consider uncertainty analysis. Notice that, this is specific for uncertainty analysis and not valid for a general metamodeling problem.

Consider a total of M time points in the simulation, $t_1, \cdots, t_M$. At time point t_i, the output $f(\boldsymbol{x}_i, \boldsymbol{u}_0)$ is known. In practice, $\boldsymbol{u}$ usually has a high probability around its nominal value. Then it is reasonable to assume that $\boldsymbol{u}$ does not deviate far from $\boldsymbol{u}_0$, especially when the uncertainty is not large. Therefore, the outputs $f(\boldsymbol{x}_1, \boldsymbol{u}_0), \cdots, f(\boldsymbol{x}_M, \boldsymbol{u}_0)$ should have a similar trend as the outputs with a general

$\boldsymbol{u}$, $f(\boldsymbol{x}_1, \boldsymbol{u}), \cdots, f(\boldsymbol{x}_M, \boldsymbol{u})$. Based on this idea, we propose

$$\psi(f(\boldsymbol{x}, \boldsymbol{u})) = \psi(f(\boldsymbol{x}, \boldsymbol{u}_0)) + \delta(\boldsymbol{x}, \boldsymbol{u}), \tag{32}$$

where $\psi(\cdot)$ is an appropriate transformation on the response, $f(\boldsymbol{x}, \boldsymbol{u}_0)$ is the evaluation of the computer model with the uncertainty parameters at their nominal values $\boldsymbol{u}_0$ and $\delta(\boldsymbol{x}, \boldsymbol{u})$ is the function used to capture the discrepancy between $\psi(f(\boldsymbol{x}, \boldsymbol{u}))$ and $\psi(f(\boldsymbol{x}, \boldsymbol{u}_0))$. Obviously, $\delta(\boldsymbol{x}, \boldsymbol{u})$ should satisfy

$$\delta(\boldsymbol{x}, \boldsymbol{u}_0) = 0, \text{ for all } x. \tag{33}$$

The reason why we include a transformation in the additive formulation (32) is that we hope the discrepancy term could be considerably small after an appropriate transformation on the response.

Since a base function is utilized in our model and the design within the local space $\boldsymbol{\mathcal{X}}_0$ is used for fitting the model, we name the model (32) "local base emulator". Local base emulator could have different forms by using different formulations for the discrepancy term. In the following two subsections, we will implement the local base emulator with linear function and GP for the discrepancy term, respectively.

3.4.1 Linear Base Emulator

If a linear function is used for the discrepancy, we call the emulator linear base emulator (LBE). Consider a linear function of $\boldsymbol{u}$ for $\delta(\boldsymbol{x}, \boldsymbol{u})$. To make the discrepancy function satisfy the constraint condition in (33), we propose

$$\delta(\boldsymbol{x}, \boldsymbol{u}) = \sum_{k=1}^{q} \beta_k(\boldsymbol{x})(u_k - u_{0k}), \tag{34}$$

where β_k is a function $\boldsymbol{x}$. Notice that if β_k is independent of $\boldsymbol{x}$, (32) becomes

$$\psi(f(\boldsymbol{x}, \boldsymbol{u})) = \psi(f(\boldsymbol{x}, \boldsymbol{u}_0)) + \sum_{k=1}^{q} \beta_k(u_k - u_{0k}).$$

To include the interactions between $\boldsymbol{x}$ and $\boldsymbol{u}$, we can further assume β_k to be a

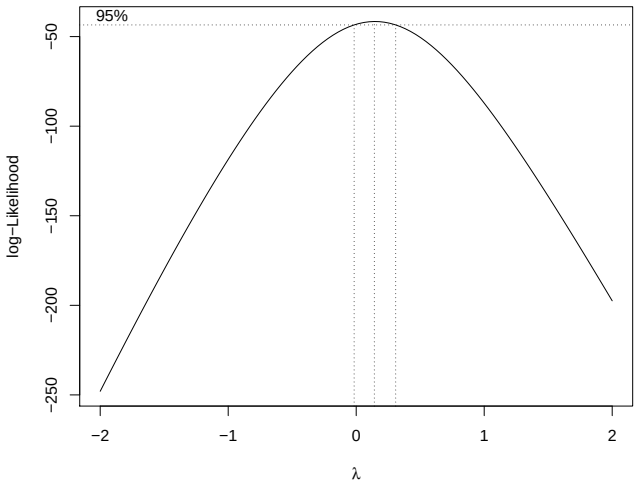

Figure 29: Log-likelihood plots for the Box-Cox transformation of tangential cutting force

linear function of $\boldsymbol{x}$. Then we have

$$\begin{aligned}\delta(\boldsymbol{x},\boldsymbol{u}) &= \sum_{j=1}^{q}(\beta_j+\sum_{i=1}^{p}\beta_{ij}x_i)(u_j-u_{0j}) \\ &= -\sum_{j=1}^{q}(\beta_j+\sum_{i=1}^{p}\beta_{ij}x_i)u_{0j}+\sum_{j=1}^{q}\beta_j u_j+\sum_{i=1}^{p}\sum_{j=1}^{q}\beta_{ij}x_i u_j. \qquad (35)\end{aligned}$$

Substituting (35) into (32), we can obtain the LBE as

$$\psi(f(\boldsymbol{x},\boldsymbol{u}))=\psi(f(\boldsymbol{x},\boldsymbol{u}_0))-\sum_{j=1}^{q}(\beta_j+\sum_{i=1}^{p}\beta_{ij}x_i)u_{0j}+\sum_{j=1}^{q}\beta_j u_j+\sum_{i=1}^{p}\sum_{j=1}^{q}\beta_{ij}x_i u_j. \quad (36)$$

Let $\boldsymbol{\beta}=(\beta_1,\cdots,\beta_q)'$ and $\boldsymbol{B}$ be a $p\times q$ matrix with ijth element β_{ij}. Since (36) is a linear model, we can easily estimate $\boldsymbol{\beta}$ and $\boldsymbol{B}$ using least squares. Denote the estimates of $\boldsymbol{\beta}$ and $\boldsymbol{B}$ by $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{B}}$. Then, the LBE is given by

$$\hat{f}(\boldsymbol{x},\boldsymbol{u})=\psi^{-1}\left(\psi(f(\boldsymbol{x},\boldsymbol{u}_0))+\sum_{j=1}^{q}(\hat{\beta}_j+\sum_{i=1}^{p}\hat{\beta}_{ij}x_i)(u_j-u_{0j})\right). \qquad (37)$$

Since (36) is a linear model, we can use a factorial design or orthogonal array to fit the model. For the solid end milling example, we use a 96-run cross array design which is a product of a 12-run orthogonal array for $\boldsymbol{x}$ and an 8-run full factorial design for $\boldsymbol{u}$.

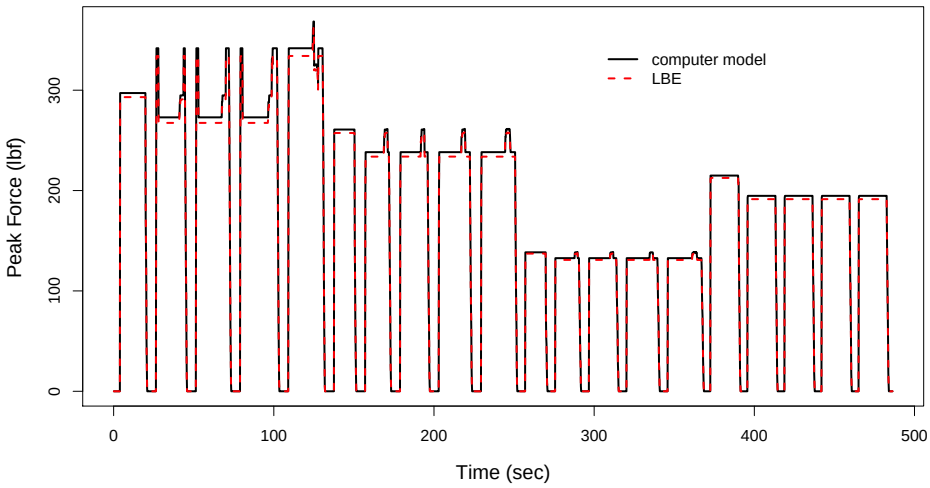

Figure 30: The predictions of LBE at a random selected $\boldsymbol{u}$

The transformation $\psi(\cdot)$ need to be determined before estimating the parameters. For a machining process, log-transformation is commonly used for the output, because the cutting force can usually be approximated by a multiplicative functional form of input variables. We also use the Box-Cox method to test the transformation on the response. The plot of log-likelihood function is shown in Figure 29. In Figure 29, we can see that $\lambda = 0$ is within 95% confidence interval, which indicates that log-transformation is appropriate for the tangential force in the solid end milling example.

The estimates of coefficient parameters $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{B}}$ are then obtained using least squares estimation. For the fitted linear model, we have an R^2 of 99.97%. By plugging in these values, we can predict the tangential force with the LBE (37).

The predictions are compared with the outputs of computer model in Figure 30 at a random set of $\boldsymbol{u}$. There is a huge improvement from kriging model, but small discrepancies still exist between the predictions and the outputs of computer model. Now we carry out the uncertainty analysis by using LBE in place of the computer model in QMC method. The 95% confidence intervals obtained from LBE and QMC are compared in Figure 31. We can see that their lower bounds agree well but some deviations are between the upper bounds. To improve the accuracy, we consider a more complex function for the discrepancy, which will be discussed in the next section.

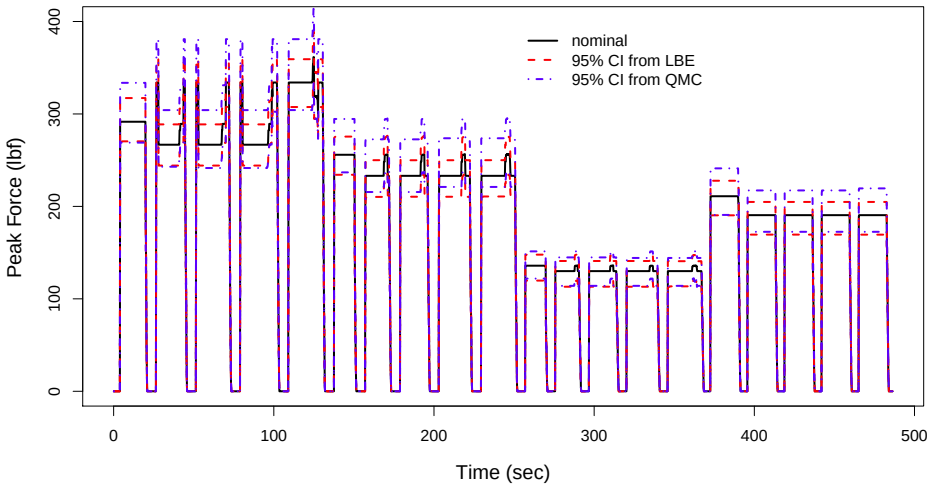

Figure 31: The simulation at nominal value of parameters and the 95% confidence interval obtained using LBE

3.4.2 Gaussian Process Base Emulator

In the previous section, we use a linear function for the discrepancy. In practice, if the performance of LBE provides a good accuracy, then we can use it for uncertainty analysis. Otherwise, we need to choose a more sophisticated functional form to capture the complex characteristics of the discrepancy.

Using two GPs, we can formulate the discrepancy as

$$\delta(\boldsymbol{x}, \boldsymbol{u}) = Z(\boldsymbol{x}, \boldsymbol{u}) - Z(\boldsymbol{x}, \boldsymbol{u}_0). \tag{38}$$

The emulators using (38) for the discrepancy term is termed as Gaussian process base emulator (GPBE). Apparently, the discrepancy function defined in (38) satisfies the constraint (33). Since both $Z(\boldsymbol{x}, \boldsymbol{u})$ and $Z(\boldsymbol{x}, \boldsymbol{u}_0)$ are GPs, $\delta(\boldsymbol{x}, \boldsymbol{u})$ is also a GP. Its mean is 0 and correlation function is given by

$$\begin{aligned} R_\delta(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{u}_i - \boldsymbol{u}_j) &= corr\left(\delta(\boldsymbol{x}_i, \boldsymbol{u}_i), \delta(\boldsymbol{x}_j, \boldsymbol{u}_j)\right) \\ &= corr\left(Z(\boldsymbol{x}_i, \boldsymbol{u}_i) - Z(\boldsymbol{x}_i, \boldsymbol{u}_0), Z(\boldsymbol{x}_j, \boldsymbol{u}_j) - Z(\boldsymbol{x}_j, \boldsymbol{u}_0)\right), \\ &= R_{\boldsymbol{x}}(\boldsymbol{x}_i - \boldsymbol{x}_j)[R_{\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_j) - R_{\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_0) - \\ &\quad R_{\boldsymbol{u}}(\boldsymbol{u}_j - \boldsymbol{u}_0) + 1], \end{aligned} \tag{39}$$

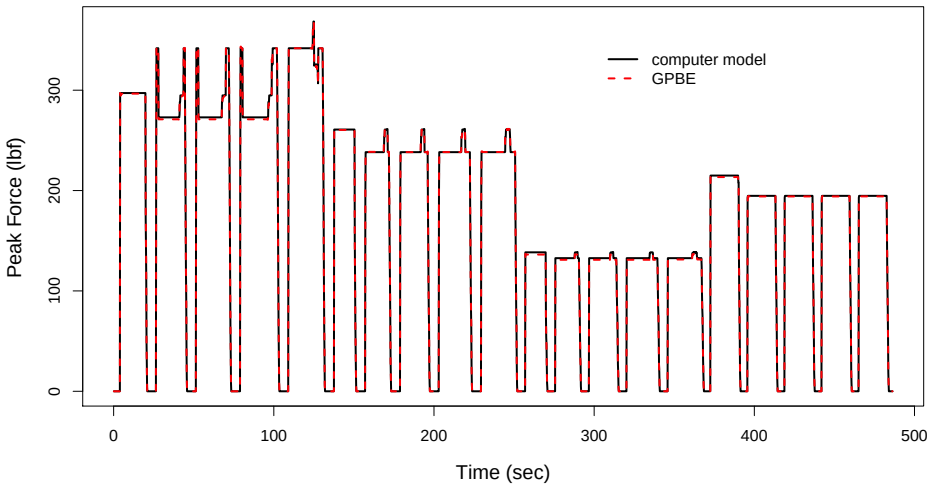

Figure 32: Comparison between the predictions of GPBE, ordinary kriging and computer model

where the correlation functions $R_{\boldsymbol{x}}$ and $R_{\boldsymbol{u}}$ have the same definition as for the ordinary kriging model. We define the vector $\boldsymbol{y}_0 = (y_{01}, \cdots, y_{0N})'$ with the ith element $y_{0i} = f(\boldsymbol{x}_i, \boldsymbol{u}_0)$, where $\boldsymbol{x}_i$ is from the ith run of the design $\boldsymbol{D}_{\boldsymbol{x},\boldsymbol{u}}$. Denote $\boldsymbol{w}$ be an $N \times 1$ vector with ith element $w_i = \psi(y_i)$, and $\boldsymbol{w}_0$ be the vector with ith element $w_i = \psi(y_{0i})$. The correlation parameters $(\boldsymbol{\theta}', \boldsymbol{\phi}')'$ can be obtained by minimizing the negative log-likelihood function

$$N \log \hat{\sigma}^2 + \log |\boldsymbol{R}_\delta|, \tag{40}$$

where $\boldsymbol{R}_\delta$ is an $N \times N$ matrix with ijth element $R_\delta(\boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{u}_i - \boldsymbol{u}_j)$ and $\hat{\sigma}^2 = (\boldsymbol{w} - \boldsymbol{w}_0)' \boldsymbol{R}_\delta^{-1} (\boldsymbol{w} - \boldsymbol{w}_0)/N$. The predictor for discrepancy function is given by

$$\hat{\delta}(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{r}_\delta(\boldsymbol{x}, \boldsymbol{u})' \boldsymbol{R}_\delta^{-1} (\boldsymbol{w} - \boldsymbol{w}_0), \tag{41}$$

where $\boldsymbol{r}_\delta(\boldsymbol{x}, \boldsymbol{u}) = (R_\delta(\boldsymbol{x} - \boldsymbol{x}_1, \boldsymbol{u} - \boldsymbol{u}_1), \cdots, R_\delta(\boldsymbol{x} - \boldsymbol{x}_N, \boldsymbol{u} - \boldsymbol{u}_N))'$. Substituting (41) into (32), we can obtain the predictor of GPBE as

$$\hat{f}(\boldsymbol{x}, \boldsymbol{u}) = \psi^{-1} \left[\psi(f(\boldsymbol{x}, \boldsymbol{u}_0)) + \boldsymbol{r}_\delta(\boldsymbol{x}, \boldsymbol{u})' \boldsymbol{R}_\delta^{-1} (\boldsymbol{w} - \boldsymbol{w}_0) \right]. \tag{42}$$

As we have validated, the log-transformation is appropriate for peak tangential force in the solid end milling example. To compare the performance of GPBE with

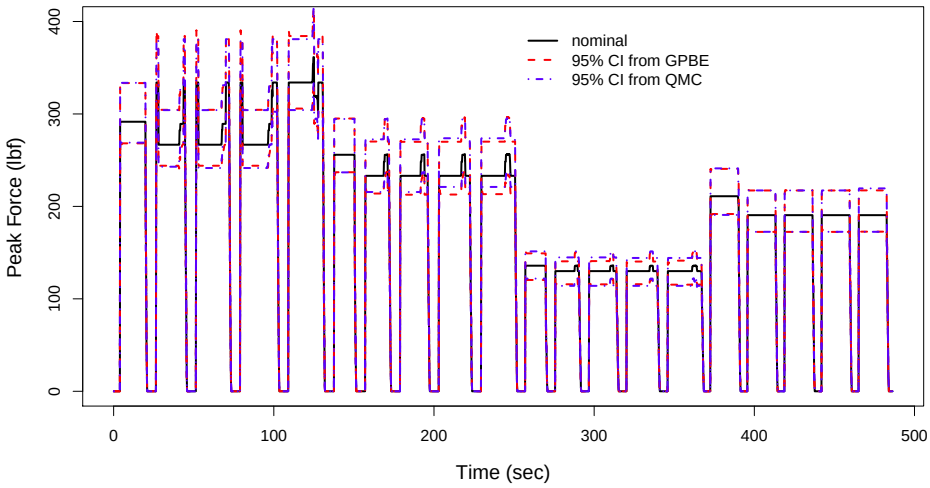

Figure 33: The simulation at nominal value of parameters and the 95% confidence interval using GPBE.

the metamodeling approach, we use the same cross array design which was used for fitting the ordinary kriging model. Similar to the inverse of correlation matrix in ordinary kriging model, we have $\boldsymbol{R}_{\delta}^{-1} = \boldsymbol{R}_{\boldsymbol{x}}^{-1} \otimes \boldsymbol{R}_{\boldsymbol{u}0}^{-1}$, where $\boldsymbol{R}_{\boldsymbol{u}0}$ is an n-by-n matrix with ijth element $R_{\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_j) - R_{\boldsymbol{u}}(\boldsymbol{u}_i - \boldsymbol{u}_0) - R_{\boldsymbol{u}}(\boldsymbol{u}_j - \boldsymbol{u}_0) + 1$. This means fitting GPBE has the same order of computational complexity as fitting an ordinary kriging model. By minimizing the negative log-likelihood function in (40), we obtain $\hat{\boldsymbol{\theta}} = (0.083, 2.052, 4.83)'$ and $\hat{\boldsymbol{\phi}} = (27.18, 0.957, 10.960, 1.005, 30.07)'$. Plugging-in these values, we have the GPBE for the solid end milling process

$$\hat{f}(\boldsymbol{x}, \boldsymbol{u}) = f(\boldsymbol{x}, \boldsymbol{u}_0) \exp\left[\boldsymbol{r}_{\delta}(\boldsymbol{x}, \boldsymbol{u})' \boldsymbol{R}_{\delta}^{-1} (\boldsymbol{w} - \boldsymbol{w}_0)\right],$$

for $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_0$. In Figure 32, the predictions of peak tangential force from GPBE are compared with the outputs of computer model at the same set of $\boldsymbol{u}$ as in Figure 30. We can see that the GPBE agrees very well with the computer model in the whole time span. Even LBE also shows a good prediction (see Figure 30), we still find GPBE achieves an additional 80% improvement with respect to the mean squared error.

Again, using GPBE in place of the computer model, we obtain the 95% confidence interval. The 95% confidence intervals from GPBE and QMC are compared in

Table 4: Comparsion of different methods for the uncertainty analysis in solid end milling process

	MC	QMC	Ordinary kriging method	LBE	GPBE
Computation time	NA	15 minutes	3 minutes	< 1s	3 minutes
Accuracy	NA	very good	poor	good	excellent

Figure 33. It shows a good agreement between the confidence intervals from the two approaches.

3.5 Conclusions

This chapter studied the uncertainty propagation in the computer model of solid end milling process. We could not solve this problem efficiently by using MC and QMC methods. Metamodel method is then considered. To obtain a good prediction for $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_0$, we generated a modified MaxPro design within the local space $\boldsymbol{\mathcal{X}}_0$ and fitted a kriging model from this design. But the fitted kriging model performs poorly in prediction. We then proposed a "local base emulator" to improve the prediction. In the local base emulator, a base function is used for capturing the general trend of the output and a discrepancy function for the remaining characteristics. We found even using a linear function for the discrepancy, we can achieve much better and faster predictions than kriging model. Using GP for the discrepancy function can further improve the prediction performance.

In Table 4, we compare the computation time and accuracy of the five methods mentioned in this chapter. QMC can provide accurate results but is slow in computation. The ordinary kriging model method is faster but poor in prediction. LBE achieves a high efficiency and a good accuracy. If we need a better accuracy, GPBE will be a desirable option. It provides an excellent prediction performance with an acceptable computational expense.

In summary, we developed a methodology which can be used for solving general uncertainty propagation problems. The advantage of the local base emulator over the other traditional approaches is that it reduces the computational expense while maintaining a good accuracy. LBE and GPBE are two options for the users. LBE

can be fast implemented and is a good option if the discrepancy can be well explained by a linear function. GPBE could be used for a more complex system.

REFERENCES

Aceil, S. M., 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.

Ba, S. and Joseph, V. R. (2012), "Composite Gaussian Process Models for Emulating Expensive Functions," *Annals of Applied Statistics*, 6, 1838-1860.

Bates, D. M., and Watts, D. G. (1988), *Nonlinear Regression Analysis and Its Applications*, New York: Wiley.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J., Lin, C. H., Tu, J. (2007), "A Framework for Validation of Computer Models," *Technometrics*, 49, 138–154.

Bondarenko V. E., Szigeti G. P., Bett G. C., Kim S. J. and Rasmusson R. L. (2004), "Computer model of action potential of mouse ventricular myocytes," *American Journal of Physiology C Heart and Circulatory Physiology* 287, H1378–H1403.

Box, G. E. P. and Hunter, W. G. (1962), "A Useful Method for Model-Building," *Technometrics* 4, 301–318.

Chang, C.-J., and Joseph, V. R. (2014),"Model Calibration through Minimal Adjustments," *Technometrics*, to appear.

Chen, W., Jin, R., and Sudjianto, A. (2004), "Analytical Uncertainty Propagation via Metamodels in Simulation-Based Design Under Uncertainty," *Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 1, 670–682.

Chen, W., Jin, R., and Sudjianto, A. (2005), "Analytical Variance-Based Global Sensitivity Analysis in Simulation-Based Design Under Uncertainty," *ASME Journal of Mechanical Design*, 127, 875–886.

Courtemanche M., Ramirez R. J., Nattel S. (1998), "Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model," *American Journal of Physiology*, 44, H301–H321.

Cryer, S. A. and Havens, P. L. (1999) "Regional sensitivity analysis using a fractional factorial method for the USDA model GLEAMS," *Environ Modell Software* 14, 613–624.

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

Dettinger, M.D.; 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

Dewettinck, K., Visscher, A., Deroo L., Huyghebaert, A. (1999), "Modeling the Steady-State Thermodynamic Operation Point of Top-Spray Fluidized Bed Processing," *Journal of Food Engineering*, 39, 131–143.

Du D., Yang H., Norring S., and Bennett E. S. (2013), "Computer modeling of glycosylation modulation dynamics in hERG channels and cardiac electrical signaling." *IEEE Journal of Biomedical and Health Informatics*.

Ednie A. R. and Bennett E. S. (2012), "Modulation of voltage-gated ion channels by sialylation," *Compr Physiol* 2(2), 1269–301.

Ednie A. R., Horton K. K., Wu J., and Bennett E. S. (2013), "Expression of the sialyltransferase, ST3Gal4, impacts cardiac voltage-gated sodium channel activity, refractory period and ventricular conduction," *Journal of Molecular and Cellular Cardiology* 59, 117–27.

Engel, R. E., Sorensen J. M., May, R. S., Doran, K. J., Trikouros, N.G., and Mozias, E. S. (1991), "Response surface development using RETRAN," *Nuclear Technol* 93, 65–81.

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.

Gilbert, P. (2012), *numDeriv: Accurate Numerical Derivatives*, R Package Version 2012.3–1.

Haaland, B. and Qian, P. Z. G. (2011), "Accurate emulators for large-scale computer experiments," *Annals of Statistics*, 39, 2974–3002.

Hamada, M. and Wu, C. F. J. (1992), "Analysis of Designed Experiments with Complex Aliasing," *Journal of Quality Technology*, 24, 130–137.

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, 23–69

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004), "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal of Scientific Computing*, 26, 448–466.

Hodgkin A. L., Huxley A. F. (1952), "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, 117, 500–544.

Hund T. J. and Rudy Y. (2004), "Rate dependence and regulation of action potential and calcium transient in a canine cardiac ventricular cell model," *Circulation* 110, 3168–3174.

Hung, Y., Joseph, V. R., and Melkote, S. N. (2014), "Analysis of Computer Experiments with Functional Response," *Technometrics*, to appear.

Joseph, V. R. (2006). "A Bayesian Approach to the Design and Analysis of Fractionated Experiments," *Technometrics* 48, 219–229.

Joseph, V. R. and Melkote, S. N. (2009), "Statistical Adjustments to Engineering Models," *Journal of Quality Technology*, 41, 362–375.

Joseph, V. R., Gul, E., and Ba, S. (2014). "A Maximum Projection Designs for Computer Experiments," submitted.

Kennedy, M. C. and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models (with discussion)," *Journal of Royal Statistical Society - Series B*, 63, 425–464.

Kim, H. K., Lee Y. W., Kim T. W., 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.

Kleijnen, J. P. C., van Ham, G., and Rotmans, J. (1992), "Techniques for sensitivity analysis of simulation models: a case study of the CO2 Greenhouse effect," *Simulation* 58, 410–417.

Kumar, M., Chang, C.-J., Melkote, S. N., and Joseph, V. R. (2013), "Modeling and Analysis of Forces in Laser Assisted Micro Milling," *ASME Journal of Manufacturing Science and Engineering*, 135, 041018.

Lee, S. H., Kim J.S., 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.

Lewins, J, Becker, M. (1982), "Sensitivity and uncertainty analysis of reactor performance parameters," *Advances in nuclear science and technology*, 14, New York: Plenum Press.

Lin, C. D., Bingham, D., Sitter, R. R., and Tang, B. (2010), "A New and Flexible Method for Constructing Designs for Computer Experiments," *Annals of Statistics*, 38, 1460–1477.

Mathelin, L. and Yousuff Hussaini, M. (2003), "A stochastic collocation algorithm for uncertainty analysis," *NASA/CR-2003-212153*

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

Noble D., Varghese A., KOHL P. and Noble P. (1998). "Improved guinea-pig ventricular cell model incorporating a diadic space, IKr and IKs, and length- and tension dependent processes," *Canadian Journal of Cardiology* 14, 123–134.

Oakley, J. E. and O'Hagan, A. (2004), "Probabilistic Sensitivity Analysis of Complex Computer Models: A Bayesian Approach," *Journal of Royal Statistical Society - Series B*, 66, 751-769.

Pal, B. K. and Joseph, V. R. (1998), "Developing Objective Strategies for Monitoring Multi-Input/Single-Output Chemical Process," *Statistical Methods for Quality Improvement* (Ed. Abraham, B.), Boston: Birkhauser, 151–162.

Qian, Z. and Wu, C. F. J. (2005), "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments," *Technometrics*, 50, 192–204.

Qian, P. Z. G., Tang, B., and Wu, C. F. J. (2009), "Nested Space-Filling Designs for Computer Experiments with Two Levels of Accuracy," *Statistica Sinica*, 19, 287–300.

Reese, C. S., Wilson, A. G., Hamada, M., Martz, H. F., and Ryan, K. J. (2004), "Integrated Analysis of Computer and Physical Experiments," *Technometrics*, 46, 153–164.

Rasmusson R. L., Clark J. W., Giles W.R., Shibata E.F., and Campbell D.L. (1990), "A mathematical model of a bullfrog cardiac pacemaker cell," *American Journal of Physiology - Heart and Circulatory Physiology*, 259, H352–H369.

Rabitz, H, Kramer, M., Dacol, D. (1983), "Sensitivity analysis in chemical kinetics," *Annual review of physical chemistry*, 34, 419-C461.

Rao, GP and Sarkar, P. K. (1997), "Sensitivity studies of air scattered neutron dose from particle accelerators," *J Stat Comput Simul* 57, 261–270.

Reese, C. S., Wilson, A. G., Hamada, M., Martz, H. F., and Ryan, K. J. (2004), "Integrated Analysis of Computer and Physical Experiments," *Technometrics*, 46, 153–164.

Ronen, Y. (1988) "Uncertainty analysis", *Boca Raton*, FL: CRC Press.

Rudy, Y. and Silva, J. R. (2006), "Computational biology in the study of cardiac ion channels and cell electrophysiology," *Quarterly Reviews of Biophysics* 39, 57–116.

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

Saltelli, A. (2002), "Making Best Use of Model Evaluations to Compute Sensitivity Indices," *Computer Physics Communication*, 145, 280-297.

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

Sobol, I. M. (2001), "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates," *Mathematics and Computers in Simulation*, 55, 271-280.

Singh, R. K. and Melkote, S. N. (2009), "Force Modeling in Laser Assisted Micro-Grooving Including the Effect of Machine Deflection," *ASME Journal of Manufacturing Science and Engineering*, 131, 0110131–0110139.

Singh, R. K., Joseph, V. R., and Melkote, S. N. (2011), "A Statistical Approach to the Optimization of a Laser-Assisted Micromachining Process," *International Journal of Advanced Manufacturing Technology*, 53, 221–230.

Tang, D.H.; Pinder, G.F. (1977),"Simulation of groundwater flow and mass transport," *Advances in Water Resources* 1, 2–30

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

Turányi, T. (1990), "Sensitivity analysis of complex kinetic systems, tools and applications," *J Math Chem* 5, 203–248.

Xu H., Guo W., and Nerbonne J. M. (1999), "Four kinetically distinct depolarization-activated K^+ currents in adult mouse ventricular myocytes," *The Journal of General Physiology* 113, 661–677.

Wang, S., Chen, W., and Tsui, K.-L. (2009), "Bayesian Validation of Computer Models," *Technometrics*, 51, 439–451.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25.

Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, M., and Keller-MacNulty, S. (2006), "Combining Experimental Data and Computer Simulations, With and Application to Flyer Plate Experiments," *Bayesian Analysis*, 1, 765-792.

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

Zeitoun, D. G., Braester, C. (1991), "A Neumann expansion approach to flow through heterogeneous formations Stochastic Hydrology and Hydraulics," *Stochastic Hydrology and Hydraulics*, 5, 207–226