STATISTICAL VALIDATION AND CALIBRATION OF COMPUTER MODELS

A Thesis
Presented to
The Academic Faculty

by

Xuyuan Liu

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

Georgia Institute of Technology
May 2011

STATISTICAL VALIDATION AND CALIBRATION OF COMPUTER MODELS

Approved by:

Professor Kwok-Leung Tsui, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor David M. Goldsman
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Ying Hung
Department of Statistics
Rutgers University

Professor Jianjun Shi
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Roshan Joseph Vengazhiyil
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: January 6, 2011

To my parents,

and my girlfriend Ruosha.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my special and sincere thanks to my dissertation advisor, Dr. Kwok-Leung Tsui, for his inspiration, guidance, and encouragement during my studies in the School of Industrial and Systems Engineering at Georgia Institute of Technology.

I would also like to express my great appreciation to Dr. David Goldsman, Dr. Ying Hung, Dr. Jianjun Shi and Dr. Roshan Joseph Vengazhiyil for serving on my dissertation committee and for their valuable suggestions and comments. I am very grateful to Dr. Ying Hung for her continued support, valuable discussions, and critical comments throughout my research.

I would like to extend my appreciation to all my friends at the Georgia Institute of Technology for their continued help and support.

Finally, I would like to thank my parents and my girlfriend Ruosha. Their love, support, encouragement and tolerance have helped me through difficult times and given me strength and courage to face challenges.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis deals with modeling, validation and calibration problems in experiments of computer models. Computer models are mathematic representations of real systems developed for understanding and investigating the systems. Before a computer model is used, it often needs to be validated by comparing the computer outputs with physical observations and calibrated by adjusting internal model parameters in order to improve the agreement between the computer outputs and physical observations. As computer models become more powerful and popular, the complexity of input and output data raises new computational challenges and stimulates the development of novel statistical modeling methods.

One challenge is to deal with computer models with random inputs (random effects). This kind of computer models is very common in engineering applications. For example, in a thermal experiment in the Sandia National Lab (Dowding et al. 2008), the volumetric heat capacity and thermal conductivity are random input variables. If input variables are randomly sampled from particular distributions with unknown parameters, the existing methods in the literature are not directly applicable. The reason is that integration over the random variable distribution is needed for the joint likelihood and the integration cannot always be expressed in a closed form. In this research, we propose a new approach which combines the nonlinear mixed effects model and the Gaussian process model (Kriging model). Different model formulations are also studied to have an better understanding of validation and calibration activities by using the thermal problem.

Another challenge comes from computer models with functional outputs. While many methods have been developed for modeling computer experiments with single response, the literature on modeling computer experiments with functional response is sketchy. Dimension reduction techniques can be used to overcome the complexity problem of function response; however, they generally involve two steps. Models are first fit at each individual setting of the input to reduce the dimensionality of the functional data. Then the estimated parameters of the models are treated as new responses, which are further modeled for prediction. Alternatively, pointwise models are first constructed at each time point and then functional curves are fit to the parameter estimates obtained from the fitted models. In this research, we first propose a functional regression model to relate functional responses to both design and time variables in one single step. Secondly, we propose a functional kriging model which uses variable selection methods by imposing a penalty function. we show that the proposed model performs better than dimension reduction based approaches and the kriging model without regularization. In addition, non-asymptotic theoretical bounds on the estimation error are presented.

CHAPTER I

INTRODUCTION

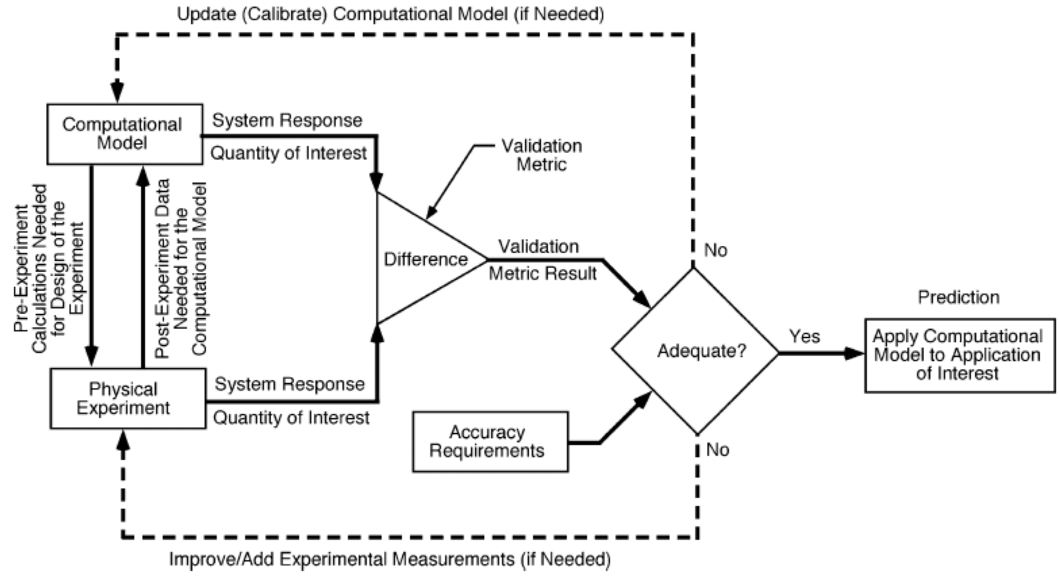
1.1 Overview

Over the past decade, the use of complex computer models has grown tremendously in modern science and engineering, and there are increasingly larger demands on the computer experiments. Computer codes are being developed to deal with weather modeling, chemical and biochemical reactions, particle physics, cosmology, semiconductor design, aircraft design, automotive crash simulations, and much more. As the power of computers increases, the scientific community has relied more and more heavily on these models. They are used for a variety of tasks, including parameter studies, design, and forecasting, and model predictions are often used to support high-consequence decisions. Computer models are often much less expensive to run than physical experiments, and in many cases, it is not possible at all to conduct physical experiments.

Statistical analysis of computer models faces the following challenges. First, we can only observe outputs at a limited number of choices for the input due to expensive running time. Furthermore, the number of input variables may be so large that a systematic exploration of all possible input combinations of interest may not be possible. Second, the values for some of the input variables may be unknown or randomly distributed for the real physical process. Physical experiments are often conducted to calibrate those unknowns. Third, the computer models are never completely accurate representations of the real processes being modeled. The model inadequacy or model uncertainty should be incorporated in statistical analysis of computer experiments.

This dissertation proposal is motivated by two challenging problems recently arising in the scietific community: functional outputs produced by computer models and uncertainty in the computer model inputs. By combining the regression-type model and the Gaussian process model, we develop different approaches to address those challenges and illustrate them by two real applications.

1.2 A Framework for Computer Experiment



Update (Calibrate) Computational Model (if Needed)
Computational Model
System Response
Quantity of Interest
Validation Metric
Pre-Experiment Calculations Needed for Design of the Experiment
Post-Experiment Data Needed for the Computational Model
Difference
Validation
Metric Result
No
Prediction
Adequate?
Yes
Apply Computational Model to Application of Interest
Physical Experiment
System Response
Quantity of Interest
Accuracy Requirements
No
Improve/Add Experimental Measurements (if Needed)

Figure 1.1: A framework for computer experiment (Oberkampf et al. (2006))

Computer experiments can be considered as equivalent to physical experiments, but performed on the computer models which represent physical processes. Physical processes are complex and as a result the codes that represent them are also complex. The complexity of computer codes makes them costly in terms of run times and this results in small output data sets. It makes design of computer experiment an important issue since you are given limited time and resource.

There are roughly five stages that go into developing a computer experiment. These are as follows:

- Formulation of the problem and identification of inputs to a computer model
- Function implementation with computer codes
- Derivation of inputs levels or the design and its application to the codes to obtain output
- Calibration and validation of the model with physical data
- Application of the results from the code to meet engineering goals

The first two stages usually involve engineers and model designers only. In this thesis, we place our focus on the last three stages which are summarized in Figure 1.1.

1.3 Statistical Analysis Issues on Computer Experiments

Statistical principles have been actively involved in the study of computer models, especially in the following areas.

- **Design of computer experiments** - The goal is to choose $x_1, \ldots, x_n$, the input values at which the computer model will be exercised. Classical design of experiments techniques, such as replication, randomization, or blocking do not apply, since what we are trying to predict is deterministic computer output with no observational error. McKay et al. (1979) first introduced Latin hypercube sampling to design computer experiments. The intuition is to cover the range of the key input values and fill the space effectively (Sacks et al. (1989); Bates et al. (1996)).
- **Surrogate model and prediction** - This is the subject of a specialized field in statistics that started with the seminal paper (Sacks, Welch, Mitchell, and Wynn, 1989) with the title "Design and Analysis of Computer Experiments".

The objective is to predict the computer model output $y^M(z)$, at any untried input z, conditional on the computer model runs at the design points $y^M(x_1), \ldots, y^M(x_n)$. The resulting prediction, together with the associated uncertainty, is then used as a fast surrogate to the computer model, often called an emulator. Since prediction may be many orders of magnitudes faster than running the computer model code itself, the emulator may eventually replace the computer model. In the following chapter, a detailed description will be given on this issue.

- **Validation** - The process of determining in what degree a computer model accurately represents the real system is referred to as model validation (AIAA G-077-1998) that generally involves the comparison of outputs computed from a computer model to observations collected from physical experiments. One approach for validating computer models is to formulate model validation as a hypothesis testing problem (Hills and Trucano 1999, Hills and Trucano 2002, Hills 2006). Oberkampf and Barone (2006) argue that computer model validation should be done quantitatively through the use of computable measures that compare computer outputs and physical observations over a range of input variables. Those measures have been referred to as validation metrics.

- **Calibration** - A simple explanation is to adjust a set of parameters associated with a computational science and engineering code so that the model agreement is maximized with respect to a set of experimental data (Trucano et al. (2006)). The most common example of a calibration method used in practice is linear or nonlinear least-squares regression. However, those methods assume no uncertainty in the computer model itself. Recently, Kennedy and O'Hagan (2001) develop a statistical model to classify sources of uncertainty arising in the use of computer models. The Bayesian approach they proposed is the first attempt to calibrate the computer model with explicit consideration of all the sources of uncertainty .

- **Sensitivity and uncertainty analysis** - Sensitivity analysis is the study of how variation in an observed response can be apportioned to different possible sources or factors. Saltelli et al. (2000) presents the large literature on this problem. Uncertainty analysis is the process of estimating the probability distribution of the model output that is implied by probability distributions associated with model inputs.

1.4 Thesis outline

This thesis is concerned with validation of computer models with random inputs or with functional outputs. In general, given the input vector z, the computer model produces a value denoted as $y^M(z)$ or a function of time denoted as $y^M(z,t)$. In many problems, z can be written as $z=(x,\theta)$, where x is a vector of controllable inputs, θ is a vector of unknown calibration parameters that reflect key charateristics of the field runs. We use θ_{ij} to represent the true values (related with the real process), for the j^{th} tested specimen in the i^{th} configuration. The remainder of the dissertation is organized as follows. Chapter 2 gives a review of existing methods for modeling of computer outputs, model validation and calibration. Afterwards, Chapter 3 and 4 move to model validation problems with two recent challenges. In Chapter 3, we will study the validation and calibration problem with unknown random inputs. Specificly, the number of unknown θ is as many as that of the field specimens, and must be dealt with in the analysis. The existing Bayesian approach becomes computationally expensive as the number of the field specimens becomes larger since more parameters are required to update through MCMC. To overcome the computational problem, We propose a mixed effects model approach for the problem and use the MLE method to implement it. Furthermore, the proposed approach is integrated with existing Gaussian process based methodology. The major challenge in Chapter 4 is the high frequency functional response which complicates the problem by the curse of dimensionality. We first propose a functional regression model which are easy to implement and very efficient for functional outputs. In the later part of the chapter, a novel approach is developed to enhance and simplify the use of Gaussian process interpolation to functional responses.

CHAPTER II

REVIEW OF EXISTING METHODS

In this chapter, we will review some existing methods for computer experiment problems under different setups. The first section focuses on the popular surrogate modeling approach, the Gaussian process model. The second and last sections illustrate different approaches for model validation problems with and without calibration parameters respectively.

2.1 Gaussian Process Models

Often times people develop a computer model to describe the relationship among a set of input and output parameters. Such a model may serve a variety of purposes, but the use of the model itself is often constrained by time and/or cost to exhaustively explore the relationship between the inputs and outputs. In this case, people might want to develop a "surrogate model" as an inexpensive approximation of the functional relationship that is described by the computer model.

Gaussian process (GP) model (kriging model) is a powerful technique originated from spatial statistics that has recently gained interest in the engineering community for its potential as a surrogate modeling technique. GP modeling uses a set of observed inputs and outputs (the "training data"; for example the results from ten different runs of a computer model) to construct an approximation to the underlying relationship. In most cases, one wants the resulting approximation to directly interpolate the observed data (as in the case of a surrogate to a deterministic computer simulation), and GP models are typically constructed in this manner, but the flexibility does exist to construct GP models that instead "smooth" or regress the observations.

One of the primary advantages of GP interpolation is that it is a non-parametric technique, which means that a priori assumptions about the functional relationship that exists between the inputs and the outputs (e.g., a linear relationship) are not required. However, the framework is still quite flexible: assumptions about smoothness properties can be reflected in the model, and large-scale variations can be captured via a parametric trend function.

2.1.1 Model Formulation

Consider that one wants to build an approximation to a function of a d-dimensional vector-valued input $x = (x^1, \ldots, x^d)$, based only on m observations of the inputs and outputs: $Y(x_1), \ldots, Y(x_m)$. The basic idea of the GP interpolation model is that the outputs, Y , are modeled as a Gaussian process that is indexed by the inputs, x. A Gaussian process is simply a set of random variables such that any finite subset has a multivariate Gaussian distribution. A Gaussian process is defined by its mean function and covariance function, which in this case are functions of x. Once the Gaussian process is observed at m locations $x_1, \ldots, x_m$, the conditional distribution of the process can be computed at any new location, x^*, which provides both an expected value and variance (uncertainty) of the underlying function.

The key here is that the function describing the covariance among the outputs, Y , is a function of the inputs, x. The covariance function is constructed such that the covariance between two outputs is large when the corresponding inputs are close together, and the covariance between two outputs is small when the corresponding inputs are far apart. As shown below, the conditional expected value of $Y(x^*)$ is a linear combination of the observed outputs, $Y(x_1), \ldots, Y(x_m)$, in which the weights depend on how close x^* is to each of $x_1, \ldots, x_m$. In addition, the conditional variance (uncertainty) of $Y(x^*)$ is small if x^* is close to the training points and large if it is not.

To develop the theory, let $Y(x)$ denote a Gaussian process which can be written as

$$Y(x) = f^T(x)\beta + \epsilon(x) \tag{2.1}$$

where $f(x) = (f_1(x), \ldots, f_m(x))^T$ is a set of pre-specified functions and $= (\beta_1, ..., \beta_m)^T$ is a set of unknown coefficients. The $\epsilon(x)$ is assumed to be a realization of a stationary Gaussian process with covariance

$$cov(\epsilon(x), \epsilon(x^*)) = \sigma^2 R(x, x^*) = \sigma^2 \exp[-d(x, x^*)]. \tag{2.2}$$

The correlation function $R(x, x^*)$ in (2) is a function of the "distance" between x and x^* . Many correlation functions are available in the literature (Santner et al. (2003)). In this research, we will consider the Gaussian correlation function which implies the distance $d(x, x^*) = \sum \phi_i(x^i - (x^*)^i)^2$. In general, a flexible distance function can be defined as $d(x, x^*) = \sum_1^d \phi_i(x^i - (x^*)^i)^{\rho_i}$, where $\phi = (\phi_1, \ldots, \phi_d)$ and $\rho = (\rho_1, \ldots, \rho_d)$ are scale and power parameters, respectively.

2.1.2 Predicting Output and Predictive Distributions

In the general case, we observe $y = (y_1, \ldots, y_n)^T$ and are interested in predicting y at a new point x^*. The empirical best linear unbiased predictor (BLUP) (Santner, Williams and Notz 2003) is

$$\hat{y}(x^*) = E(y(x^*)|y) = f(x^*)^T\hat{\beta} + rR^{-1}(y - F\hat{\beta}) \tag{2.3}$$

where $r = (R(x^*, x_1), \ldots, R(x^*, x_n))^T$, $\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y$, R is the $(n \times n)$ matrix with entries $R(x_i, x_j)$ for $i, j = 1, \ldots, n$ and $F = (f(x_1)^T, \ldots, f(x_n)^T)^T$. Usually the correlation parameters ϕ are unknown and need to be estimated from data. It is done by maximizing the log likelihood

$$-\frac{1}{2}(n \ln(\hat{\sigma^2})) + \ln|R|), \tag{2.4}$$

where $\hat{\sigma^2} = \frac{1}{n}(y - F\hat{\beta})^T R^{-1}(y - F\hat{\beta})$.

Often times, a predictive distribution for $y(x^*)$ is desired since it provide very useful information for model validation and uncertainty quantification, which we will see in the next section. The predictive distribution of $y^* = y(x^*)$ based on y is defined to be the conditional distribution of y^* given y (denoted by $[y^*|y]$). It has been shown (Santner et al. (2003)) that assuming σ^2 and correlation parameters ϕ known and $\beta \sim N(b_0, \tau^2 V_0)$, the predictive distribution of y^* is a normal distribution. When only ϕ are known, assuming $\sigma^2 \sim \chi_v^2$ or $\sigma^2 \sim \frac{1}{\sigma^2}$, the predictive distribution of y^* is a t-distribution. When both σ^2 and ϕ are unknown, there are two ways to get the predictive distribution of y^*. The first is to plug in $\hat{\phi}$ which is an estimator of ϕ (usually MLE). An alternative way that accounts for uncertainty in ϕ is to assume a prior distribution for ϕ and derive the full posterior distribution $[y^*|y]$.

2.2 Model Validation

Model Validation explores the degree to which the predictive capability of the model is suitable for a particular purpose. In general, there are several sub-fields within model validation, such as the design of the validation experiments, the design of the computer experiments, uncertainty quantification, and validation (or comparison) metrics. Bayarri et al. (2007) proposed a framework for validation of computer models which consists of six steps. They involve (1) problem definition (inputs, outputs, initial uncertainties); (2) evaluation criteria; (3) experimental design; (4) computer model output approximation; (5) comparison and the combination of field and computer run data; (6) feedback for revising the model, performing additional experiments, and so on. This framework can help engineers conduct validation studies under a statistical guidance. In a narrow sense, step 5 is usually considered as the central piece of the validation activity which involves comparing and combining computer outputs with physical observations.

Thereafter, we will focus on study of validation metrics which is the basis for comparing computer outputs and physical observations and analysis of combination of computer outputs and physical observations.

The first statistical method introduced as validation metrics is hypothesis testing (Hills and Trucano 1999, Hills and Trucano 2002, Hills 2006). For example, Hills and Trucano (2002) considered a χ^2 test for computer model validation. It was assumed that the vectors of computer outputs and physical observations follow independent multivariate normal distributions and they computed a χ^2 statistic to test a null hypothesis that the model bias (i.e., the difference of the two vectors) has a zero mean.

Oberkampf and Barone (2006) gave a comprehensive review on validation metrics of computer model validation. They discuss a variety of properties that a validation metric should possess and emphasize that a validation metric should quantify uncertainties in the comparison of computer outputs and physical observations. Uncertainties could be due to random experimental errors in the physical observations or errors resulting from post-processing computer outputs and physical observations, (e.g., errors resulting from fitting models to computer outputs or physical observations). Oberkampf and Barone (2006) propose a validation metric that uses the statistical confidence intervals for quantifying the uncertainties. In their approach they first fitted a nonlinear regression model to physical observations and constructed a confidence band together with the fitted curve. They then compared the fitted regression curve and its confidence band with the computer outputs for model validation. For the regions of input space where the computer outputs fall outside the confidence band, the computer model is considered to be inadequate in those input regions. However, there are some concerns over their approach such as mis-specification of a nonlinear regression model and using only physical data to build model.

To overcome those drawbacks, Wang et al. (2009) propose a Bayesian approach for computer model validation. First, they considered fitting Gaussian processes to the data, which is non-parametric and flexible. Second, their proposed approach combines both physical observations and computer outputs to provide more accurate prediction than fitting a model using only physical observations. Finally, the Bayesian approach produces a posterior distribution of the system output, thus provides a simple way to compute the prediction intervals of the true system output. Furthermore, this approach derives the posterior distributions of the computer model and bias function (the deviation between the computer model and true outputs) separately. This allows them to decompose the prediction error and understand how combining computer outputs and physical observations can provide more accurate predictions than using only one set of data.

Recently, a new validation metric is developed by Ferson et al. (2008) to deal with the case in which either or both predictions and data are expressed as probability distributions. When the computer outputs are probability distributions due to the variability in some parameters, it becomes insufficient to only compare the mean behaviors of computer outputs and physical observations. In order to compare the entire distribution, the proposed validation metric is based on the non-parametric test statistics which is very similar to the Cramer-von-Mises test statistic. They also considered using probability distribution function to transform the observations into common uniformly distributed ones. By using such a transformation, the data over the entire validation domain can be integrated into a single measure of overall disagreement. We will further illustrate this idea in Chapter 3.

Another problem arising in model validation is the needs to deal with functional outputs, which are not only functions of physical attributes but also functions of time or space. Bayarri et.al (2005) considered a Gaussian process approach that considers time as an additional model input. However, this straightforward approach can only work when we have limited observations which is usually not the case for functional outputs. Bayarri et al. (2007) proposed a new approach that utilizes a wavelet representation of the functional data. They apply a hierarchical version of the scalar validation methodology to the wavelet coefficients, and transform back, to ultimately compare computer model output with field output. Alternatively, Hidgon et al. (2008) considered using basis vectors based on singular value decomposition (SVD) to represent the functional data. Again, the weights of those basis vectors are modeled as Gaussian processes. They apply their approach to a different objective, however, the common idea of these two approaches is that different strategies are adopted for modeling function data over physical attributes and over time or space. By doing so, the efficiency of modeling functional outputs can be greatly improved. Recently, people begin research on using the straightforward approach with subset of functional data which are carefully selected from the entire dataset. McFarland et al. (2008) proposed a point selection scheme which, in each step, selects the point with the largest prediction error based on the existing fitted model. Hung et al. (2009) consider a different approach in which a point is selected if it gives the largest improvement of overall prediction error. In addition, they also develop a new updating scheme for calculating covariance matrix which can further improve the efficiency of the overall algorithm. In Chapter 4, we will consider a modification to Hung et al.'s approach to adapt to a different situation.

2.3 Model Validation with Presence of Calibration Parameters

We can directly compare and combine computer outputs with physical observations when computer models only have controllable parameters. However, in reality, it is very common that there is uncertainty in the value of important physical constants which researchers can not control. For example, a simple chemical kinetics model (Loeppky et al. (2006)) is defined as follows:

$$\phi(x|\tau) = c + y_0 \exp(-\tau x),$$

where c is the residual concentration of the chemical at the end of the reaction process, y_0 is the initial concentration of the chemical, x is the time and τ is an unknown decay rate that is specific to the chemical reaction under consideration. It is clear that without a correct knowledge of τ, the model cannot be used for further analysis. The estimation of such a parameter via experimental observations might be termed *model calibration*.

This type of model calibration is quite widespread in quantitative analysis. One common example is that of linear regression analysis. A linear relationship is postulated between a dependent variable and one or more independent variables, and then a set of observed values are used to estimate the unknown parameters that determine the model.

However, the meaning of the term calibration, when used in reference to computer simulations, must be defined more explicitly, because in the modeling and simulation fields, the word calibration can have several interpretations. Trucano et al. (2006) define calibration as to adjust a set of code input parameters associated with one or more calculations so that the resulting agreement of the code calculations with a chosen and fixed set of experimental data is maximized. McFarland et al. (2008) consider calibration and validation both involving comparing the computational implementation of a model against experimentally observed outcomes, but the objective of calibration is to make inferences about unknown parameters that govern the computational implementation.

Trucano et al. also raised an important question about distinguishing between calibration and validation. Since both activities involve comparing the computer outputs and physical observations, there is an strong interaction between calibration and validation. Without calibration efforts, it is difficult to get satisfying validation results. However, calibration under wrong model assumptions may lead to misleading validation results. Therefore, it is very important to separate calibration and validation clearly in scientific studies.

Hereafter, we will review general calibration methods which include classical regression methods and recent developed Gaussian process approaches. Campbell (2006) gives an overview of various statistical methods that have been proposed for the calibration of computer simulations. One of the most popular approaches is to express the calibration problem in terms of nonlinear regression analysis (Trucano et al., 2006).

When nonlinear regression analysis is applied to the calibration of computer models, the dependent variable is the computer output, the independent variables are typically observable experimental conditions, and the unknowns are those internal computer model parameters that are to be estimated. Thus, the relationship between the dependent and independent variables is expressed as

$$y_i = S(x_i, \theta) + \epsilon_i$$

where y_i's is the physical response, ϵ_i is a mean zero random error, θ is a p-dimensional vector of calibration parameters, x_i is the vector of design or controllable parameters and $S(\cdot, \cdot)$ represents the computer model.

In general, we can assume that $\epsilon = (\epsilon_1, \dots, \epsilon_n)^T$ have a multivariate normal distribution $N(\mathbf{0}, \sigma^2 R)$. In the case that $R = I$, ϵ_i are taken to be independently and identically distributed. Under this assumption, the estimation of θ can be done by minimizing the weighted sum of squared errors function:

$$LS(\theta) = (y - S(x, \theta))^T R^{-1} (y - S(x, \theta)).$$

The uncertainty of the estimator $\hat{\theta}$ is characterized by the covariance matrix $cov(\hat{\theta}) = s^2(V'V)$, where V is a $n \times p$ matrix and $V_{ij} = \frac{\partial S(x_i, \theta)}{\partial \theta_j}|_{\theta = \hat{\theta}}$

The nonlinear regression methods described above are used in calibrating parameters of models, taking into account uncertainty in the physical data but assuming no uncertainty in the model itself. This motivates people to develop formal statistical methods that address model uncertainty. One important approach is that of Kennedy and O'Hagan (2001). Their formulation for calibration data includes an random error term (similar to the random error term in nonlinear regression) and a model discrepancy term, which represents the deviation of the computer model from real process. Model discrepancy is modeled by a Gaussian process.

The model can be written as

$$y^R = \rho\eta(x, \theta) + \delta(x),$$

$$y^F = y^R + \epsilon.$$

$\eta(\cdot, \cdot)$ represents the computer model, similar to $S(\cdot, \cdot)$, however, a different notation is used here since it could also be a surrogate model (but not fitted separately). ρ is introduced as an unknown regression parameter, and $\delta(x)$ is the model discrepancy or model inadequacy function and is treated as independent of $\eta(\cdot, \cdot)$.

The estimation of parameters is done by using MCMC. The posterior distribution of θ is considered as the calibration result. To predict the future output at a new site x^*, they use the posterior mean of y^F given data $(y = Y)$: $E(y(x^*)|Y) = E_\theta(E(y(x^*)|Y, \theta))$.

CHAPTER III

VALIDATION WITH THE PRESENCE OF CALIBRATION PARAMETERS UNDER MIXED-EFFECT MODELS

3.1 Introduction

We have discussed the problem of model validation with calibration parameters in Chapter 2. In all approaches introduced, though they consider the uncertainty of the calibration parameters, they all assume that the calibration parameters are held constant over all experiments. However, in reality, uncertainties can arise from a variety of sources: uncertainty in the specification of initial conditions; uncertainty in the value of important physical constants (e.g., thermal conductivity, equations of state, and material strength); inadequate mathematical models in the code to describe physical behavior; and inadequacies in the numerical algorithms used for solving the specified mathematical systems (e.g. unresolved grids).

There are two forms of uncertainty explicitly addressed in the thermal challenge problem. These include uncertainty associated with the thermal properties due to random variability in the material reflected through the thermal properties, and uncertainty due to possible model form error associated with a significant, but not modeled temperature dependence in a thermal property.

For the uncertainty related to random variability of thermal properties, it could be treated as a random effect in a standard mixed model. This motivates us to consider using well-established method for nonlinear mixed effect models on this thermal problem.

In this chapter, we describe an approach for combining observations from field experiments with the mathematical model of a physical process to carry out statistical inference. Of particular interest here is determining uncertainty in the resulting predictions. This typically involves estimation of experimental errors, calibration of distributions of parameters in the computer simulator, and accounting for inadequate physics in the simulator.

This chapter is organized as follows. We give a description of thermal challenge problem in the next section. In Section 3.3, some existing methods are reviewed and discussed. We propose our new approach and formulations in Section 3.4 and illustrate the results in Section 3.5, followed by discussion and some future works in Section 3.6 and 3.7.

3.2 Problem Description

The thermal validation challenge problem (Dowding et al., 2008), developed at Sandia National Laboratories, is a hypothetical problem that presents the analyst with several pieces of validation data and a corresponding mathematical model. In this problem, we have an ensemble of safety-critical devices that exhibit unit-to-unit variability due to manufacturing processes. Each device is a material layer of thickness L that is exposed to a heat flux. In the intended application the devices are exposed to environments that are well characterized, q = 3500 W/m^2. The thickness is L = 1.90 cm. These quantities are defined by specifications of performance and therefore have no uncertainty. It is prohibitively expensive to assess regulatory compliance through lot sample testing of the safety devices; consequently, it is expected that the validated model will play a critical role in the assessment of regulatory compliance.

Experimental data from a series of material characterization, validation, and accreditation experiments related to the mathematical model were provided by Sandia National Laboratories. People were asked to evaluate the validity of the provided

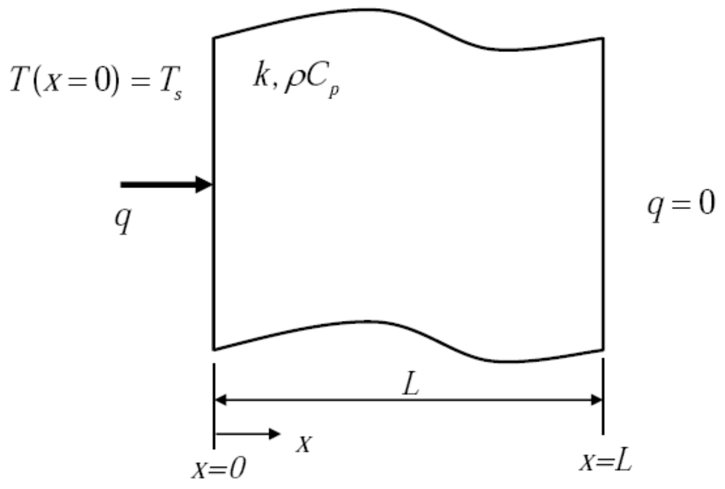


Figure 3.1: Schematic of heat conduction problem

mathematical solution for use in a specified application with a defined regulatory criterion, and were asked to use the solution to predict regulatory compliance. Following an engineering procedure, specifically, people are first expected to use material characterization data to estimate a probabilistic model for the physical properties that are inputs to the mathematical model. The second and third objectives involve assessing the model's accuracy based on available experimental data (model validation). The final objective is to use the model to predict whether or not a specified regulatory requirement will be met.

The temperature response of the device in the intended application is modeled by one-dimensional heat conduction through a slab (Fig. 1). The boundary conditions are specified flux on the x = 0 face and adiabatic on the x = L face. Furthermore, the thermal properties, κ and ρCp and initial condition, T_i, are described by constants. The analytical solution for the temperature in the body (for $t > 0$) can be written as

$$y^M(\kappa, \rho Cp, T_0, x, L, q; t)$$
$$= T_0 + \frac{qL}{\kappa}\left[\frac{\kappa t/\rho Cp}{L^2} + \frac{1}{3} - \frac{x}{L} + \frac{x^2}{2L^2} - \sum_{n=1}^{6} \frac{2}{\pi^2 n^2}\exp(-\frac{\pi^2 n^2 \kappa t}{L^2 \rho Cp})\cos(n\pi\frac{x}{L})\right]. \quad (3.1)$$

The information of all inputs of the model is summarized in Table 3.1.

Validation experiments are conducted at the four heat flux/thickness points shown

Table 3.1: The input/uncertainty table

Parameter name	Parameter symbol	Uncertainty	Current status
Thermal conductivity	κ	Yes	Unknown
Volumetric heat capacity	ρCp	Yes	Unknown
Heat flux	q	None	1000, 2000, 3000
Thickness	L	None	0.0127, 0.019, 0.254
Position	x	None	0
Initial temperature	T0	None	25
Time	t	None	0, 50, 100, ..., 1000

in Fig 3.2. The physical experiment data and simulation data based on estimated calibration parameters are presented in Fig 3.3.

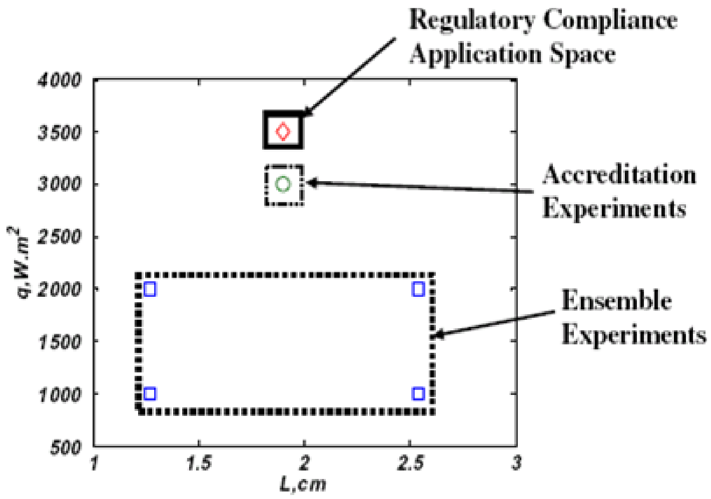


Figure 3.2: Parameter space for the validation activities. (Dowding et al. (2008))

There are some limited data with $x \neq 0$ in the accreditation data set but we ignore them because only surface temperature (x = 0) is involved in the intended application (regulatory condition) and little benefit is expected by including them. In all that follows, x is fixed at 0. We thus remove x from the input list. In the later sections, we will refer the design parameters L and q as x, and calibration parameter κ and ρCp as θ.

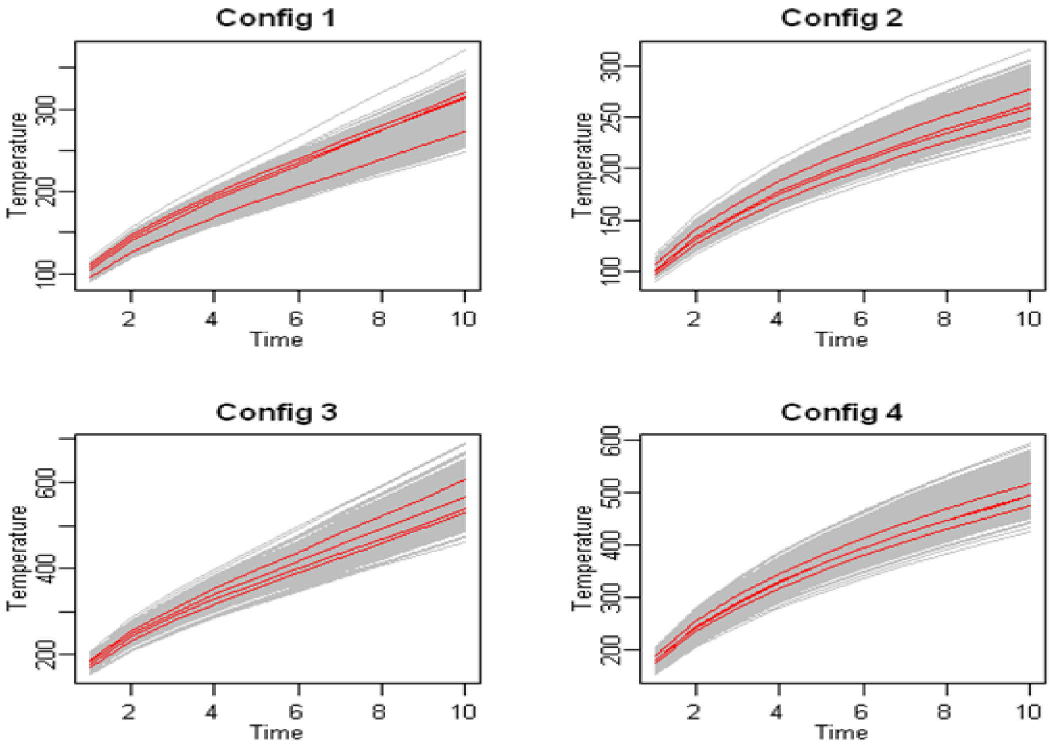


Figure 3.3: The experimental traces with simulation data: Red: Physical data; Grey: Simulation data

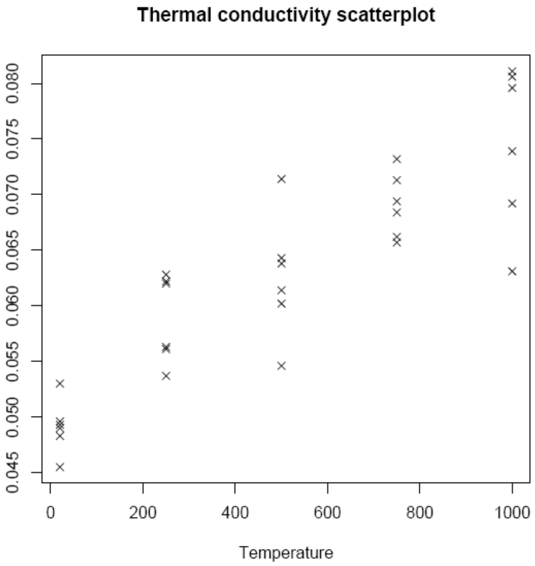


Figure 3.4: Thermal conductivity vs temperature

3.2.1 Material Characterization

The first challenge problem objective is to use data from the material characterization experiments to characterize a probabilistic model for the material properties κ and ρCp, which are inputs to the heat-transfer model given by Eq. (3.1). In a standard engineering approach, the estimated probabilistic model can be further used to generate random samples of thermal properties which will be plugged into the mathematical model for prediction. For a Bayesian approach. the estimated probabilistic model is used to be priors for MCMC implementation. Then, physical observations could be used to further calibrate the probabilistic model. Under the assumption of the thermal mathematical model, the thermal properties should be random distributed independently of other design parameters and experimental outcomes. However, this assumption is obviously violated in this problem by checking the relationship between κ and temperature (Fig 3.3). Several people (McFarland (2008), Ferson et al. (2008)) have argued that the inclusion of a temperature-dependent material model requires modification of the given analytical heat transfer solution (e.g., implementation of an iterative solution scheme), and doing so is decidedly inconsistent with the purpose of the challenge problem validation activities, which are to assess the accuracy of and make predictions with an inherently "flawed" model (one that ignores the temperature-dependence of the material properties).

However, ignorance of such dependence is problematic for calibration and prediction purpose. In this analysis, we will show that we can achieve better calibration and prediction by putting additional structures on thermal properties without changing the basic mathematical model.

3.3 Different Approaches

Hills et al. (2008) summarizes various approaches used to address the thermal validation challenge problem. In this section, we will focus on presenting some of them which are closely related to our method.

3.3.1 Gaussian Process with Bayesian Scheme

The Gaussian process model which has been described before is naturally extended to this problem. Higdon et al. (2008) and Liu et al. (2008) follow this framework. Both methods are extensions of the basic approach of Kennedy et al. (2001) in which the field observations are modeled according to the basic decomposition:

field data = simulator + discrepancy + error

where the simulator has been calibrated, or tuned, to be consistent with the field data. The discrepancy term accounts for differences between the simulation model and the physical system. This discrepancy may be due to missing or inadequate physics, as well as numerical inaccuracies in the code. A direct application of the model formulation of Kennedy et al. (2001) to the thermal problem is rather inefficient since it treats the temperature profile produced by an experiment or simulation as a sequence of separate experiments or simulations, each producing a single temperature. Because of this, the computational demands required for posterior sampling via Markov chain Monte Carlo can be prohibitive.

Facing this difficulty, Higdon et al. considered a modified approach to more efficiently deal with the multivariate nature of the output. They use basis functions to represent the multivariate output from the simulations as well as the experiments. Assume $y^M(x, \theta)$ to be a time trace of 11 points at specification (x, θ), a basis function representation is

$$y^M(x, \theta) = \sum_{i=1}^{p_m} \phi_i w_i(x, \theta) + \epsilon,$$

where $\Phi_m = [\phi_1, \ldots, \phi_{p_m}]$ is a collection of orthogonal, 11-dimensional basis vectors, the $w_i(x, \theta)$'s are GPs over input space, and ϵ is a 11-dimensional error term. Similarly, they define the discrepancy model which, like the model for $y^M(x, \theta)$, is constructed using a basis representation, placing GP models on the basis weights. Denote the discrepancy function as $\delta(x)$, then we have

$$\delta(x) = \sum_{k=1}^{p_\delta} d_k(t)v_k(x) = \sum_{k=1}^{p_\delta} d_k v_k(x), \tag{3.2}$$

where d_k's are the basis functions and $v_k(x)$'s are GPs over input space of x.

Given the model specifications for the simulator $y^M(x, \theta)$ and the discrepancy $\delta(x)$, the full model $y^F(x) = y^M(x, \theta) + \delta(x) + \epsilon$ can be rewritten as

$$y^F(x) = \Phi_m w(x, \theta) + Dv(x) + \epsilon,$$

where the discrepancy basis matrix D is determined by the function form given in (3.1). As Hidgon et al. suggest, the basis representations substantially reduce the dimension of the GP models for the emulator and the discrepancy to reduce the computational burden.

Liu et al. use another approach to overcome the diffculty with time dimension. Instead of building a surrogate model for the mathematical model, they take advantage of the computational efficiency of that model and use it directly in the full model. For the discrepancy function (or bias function), they assume a GP model on it which includes time as an additional input. Their model can be written as

$$y^F(x_i, t) = y^M(x_i, \theta_{ij}, t) + \delta(x_i, t) + \epsilon(t),$$

where $y^M(x, \theta, t)$ is the mathematical model which is given to us, $\delta(x, t)$ is a GP over input space of x and t, and $\epsilon(t)$ is the error term with autocorrelation over time.

Hidgon et al. and Liu et al. both use the Bayesian approach to implement their full models. However, in addition to the different model formulations, they have another significant difference. In Hidgon's model, though they assume random distributions on thermal properties, they also assume that thermal properties are constant over all sample materials. This constant assumption is not adopted by Liu's model where each replicate i is associated with a θ_i. Since uncertainty associated with random variability in the material is a major factor in this application, the constant assumption can't be valid. Liu's assumption is more reasonable in this application, however, the introduction of different θ_i's makes the Bayesian approach computationally very expensive. Nonetheless, their model provides a connection with likelihood-based mixed effect model which could be an interesting problem to study with in uncertainty analysis.

3.3.2 Engineering Approaches

In this subsection, we summarize two approaches from the enigeering society. Ferson et al. (2008) makes a major contribution in designing a new metric to compare data from different settings. In their modeling effort, they have concerns over the statistical modeling approach which may confound calibration with validation and prediction. Therefore, they clearly distinguish these activities in their analysis. The field data are not used to fit any model but only to provide validation metrics. The material characterization (MC) data are used to generate random samples of thermal properties which then are plugged into the mathematical model. Alternatively, they also consider to use the MC data to build a regression relationship between thermal properties and final temperature. Based on this regression relationship, they can iteratively generate random samples from the regression model and the mathematical model until the distribution of random samples converges. Ferson et al. provide an important idea to evaluate the mathematical model directly rather than using the fitted statistical model. However, ignoring the physical data entirely is not an efficient method from the statistical point of view.

Xiong et al. (2008) consider several alternative models from model updating point of view. And they also use the mathematical model directly rather than building a surrogate model for it. Therefore, their setup is very similar as Liu et al.'s. However, they implement their models using the maximum likelihood based approach rather than the Bayesian approach. They also consider incorporating the temperature dependence into one of their models and simplify the discrepancy term to be a linear function instead of a GP.

3.4 Proposed Approach

Our proposed approach is based on nonlinear regression analysis introduced in Chapter 2. There are two major contributions in our work. First, following Xiong et al's idea of model updating, we formulate their models more formally in statistical sense and implement them based on more accurate likelihood approximation from recent statistical research. Second, we use this formulation as a foundation to include Gaussian process which is more flexible than linear functions.

In general, an updated model can be formulated as follows,

$$y^F(x) = y^R(x) + \epsilon = y^M(x, \theta) + \delta(x) + \epsilon. \tag{3.3}$$

In this model formulation, x are the controllable input variables which are fixed without uncertainty. θ are the uncontrollable input variables which have to be calibrated using physical experiments. Under Kennedy and O'Hagan's assumptions, θ are assumed to be unknown but constant parameters. However, this assumption is obviously violated in the thermal problem, since the calibration parameters, θ, have unit-to-unit variability. Therefore, in our formulation, we will assume that θ follow a random distribution among different specimens. $\delta(x)$ is the bias function which is used to capture the systematic bias of computer models. There are many candidate functions for $\delta(x)$. The most popular choice is the Gaussian process, but it is also possible to consider other alternatives such as linear functions. ϵ represents random errors associated with the processes (e.g. measurement error).

3.4.1 Nonlinear Mixed-Effects Models

Here we will start with an introduction of nonlinear mixed effects model and describe in detail how to implement this model to our problem. Nonlinear mixed-effects models are mixed-effects models in which some, or all, of the fixed and random effects occur nonlinearly in the model function (Pinheiro and Bates (2000)). The basic nonlinear mixed-effects (NLME) model for repeated measures is defined as follows.

$$y_{ij} = f(\phi_{ij}, \nu_{ij}) + \epsilon_{ij}, \quad i = 1, \ldots, M, j = 1, \ldots, n_i, \tag{3.4}$$

where M is the number of groups, n_i is the number of observations on the ith group, f is a general, real-valued, differentiable function of a group-specific parameter vector ϕ_{ij} and a covariate vector ν_{ij} , and ϵ_{ij} is a normally distributed within-group error term ($\epsilon_i = (\epsilon_{i1}, \ldots, \epsilon_{i1}) \sim N(0, \sigma^2\Lambda)$). The function f is nonlinear in at least one component of the group-specific paramter vector ϕ_{ij} , which is modeled as

$$\phi_{ij} = A_{ij}\beta + B_{ij}b_i, \quad b_i \sim N(0, \Psi), \tag{3.5}$$

where β is a p-dimensional vector of *fixed effects* and b_i is a q-dimensional *random effects* vector associated with the ith group (not varying with j) with variance-covariance matrix Ψ (assuming $\Psi^{-1} = \sigma^{-2}\Delta^T\Delta$). The matrices A_{ij} and B_{ij} are of appropriate dimensions and depend on the group. Observations corresponding to different groups are assumed to be independent and the within-group errors may be heteroscedastic and correlated, but independent of the random effects b_i. In our example, for each experimental specimen, we will measure its temperatures in the experiment over time. Thus they could be thought as repeated measures and each specimen would be treated as a group. Then the mathematical model y^M or $y^M + \delta$ will be f function in the NLME model. The design variables x and the time t are ν_{ij} in the model. The calibration paramter θ and the parameter $\delta(x)$ (assuming $\delta(x) = f_\delta(x; \beta_\delta)$) are ϕ_{ij} in the model.

In general, we could assume that

$$\theta = \theta_0 + b^\theta + s_\theta(x, t; \beta_\theta), \tag{3.6}$$

where θ_0 and β_θ are fixed effects and b^θ is the random effect following a random distribution $N(0, \Psi_\theta)$. In this research, we consider two cases of s_θ: $s_\theta \equiv 0$ and $s_\theta(x, t; \beta_\theta) = z(x, t)^T \beta_\theta$. For the first case, it means that θ does not depend other design parameters or change over time. For the second case, it means just the opposite which corresponds to the temperature-dependent κ.

After having made all the assumptions and model setups, the total likelihood could be written as

$$L(\mu_\theta, \Sigma_\theta; y^F) = \iint \prod f_M(y^F(x_i)|\theta_i) f_\theta(\theta_i|\mu_\theta, \Sigma_\theta) d\tilde{\theta} \tag{3.7}$$

To maximize this likelihood, the Laplacian approximation or adaptive Gaussian approximation (Pinheiro and Bates (2000)) can be used here for approximating the loglikelihood function and a two-step alternating procedure is constructed to estimate all parameters.

1. The first step is to minimize a penalized nonlinear least squares objective function

   $$\sum_{i=1}^{M} g(\beta, \Delta, y_i, \theta_i) = \sum_{i=1}^{M} [\|\Lambda^{-T/2}[y_i - f(\theta_0 + s_\theta(x_i; \beta_\theta), b_i^\theta)]\|^2 + \|\Delta b_i^\theta\|^2]$$

   to get the conditional modes $\hat{b^\theta}$ of the random effects b^θ with $\beta = (\theta_0, \beta_\theta)$ and Δ fixed. Hereafter, we denote $f_i(\cdot)$ as $f(\theta_0 + s_\theta(x_i; \beta_\theta), b_i^\theta)$

2. The second step is to minimize the Laplacian approximation of the log-likelihood

   $$\begin{aligned} l(\beta, \Delta) &= -\tfrac{N}{2}[1 + \log(2\pi) + \log(\hat{\sigma^2})] + M \log|\Delta| \\ &\quad -\tfrac{1}{2}\textstyle\sum_{i=1}^{M} \log|G(\beta, \Delta, y_i)| - \tfrac{1}{2} M \log|\Lambda| \end{aligned}$$

   to the estimates of the fixed effects β and Δ, where

   $$G(\beta, \Delta, y_i) = \left.\frac{\partial f_i(\beta, b_i^\theta)}{\partial b_i^\theta}\right|_{\hat{b_i^\theta}} \Lambda^{-1} \left.\frac{\partial f_i(\beta, b_i^\theta)}{\partial (b_i^\theta)^T}\right|_{\hat{b_i^\theta}} + \Delta^T \Delta$$

and

$$\hat{\sigma^2} = \sum_{i=1}^{M} g(\beta, \Delta, y_i, \hat{b_i^\theta})/N.$$

These two steps will be repeated until the convergence is achieved.

3.4.2 Nonlinear mixed effects model with Gaussian process bias

The above estimation approach will only work if we assume the bias function is linear function of other parameters. In case that we want to use a Gaussian process for the bias term, the assumption of independence of between-group errors will not hold any more. Therefore we have to develop a new estimation procedure which is one of our contribution in this work. There are two ways to estimate parameters with a Gaussian process involved. The first method is still to use the above approximation but modify the objective functions to adapt to the new likelihood. The other method is to use EM algorithm to optimize the exact likelihood directly. Here, in this section, we will adopt the first method, since it is easy to implement under the current settings.

Under the Gaussian process assumption,

$$\delta(x, \theta, t) \sim N(f^\delta(x,t)^T \beta_\delta, \sigma_d^2 R)$$

where $f^\delta(x,t) = (f_1^\delta(x,t), \ldots, f_k^\delta(x,t))^T$ and R is a $N \times N$ matrix with elements

$$\begin{aligned} R((x,t,b^\theta),(x^*,t^*,b^{\theta*})) &= c((x,t,b^\theta),(x^*,t^*,b^{\theta*})) \\ &= \exp[-(\textstyle\sum_{i=1}^{p}[\phi_i^x(x_i - x_i^*)^{\rho_i^x}] + \phi_t(t-t^*)^\nu + \textstyle\sum_{i=1}^{q}[\phi_i^\theta(b_i^\theta - b_i^{\theta*})^{\rho_i^\theta}])] \end{aligned}$$

Assume that $\tilde{y} = (y_1^T, y_2^T, \ldots, y_M^T)^T$, $\tilde{f}(\beta, \tilde{b^\theta}) = (f_1(\beta, b_1^\theta), \ldots, f_M(\beta, b_M^\theta))^T$ and $\tilde{\Lambda} = \begin{pmatrix} \Lambda & 0 & \cdots & 0 \\ 0 & \Lambda & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \Lambda \end{pmatrix}$. The two-step procedure will be modified as follows.

1. The new penalized nonlinear least square objective function becomes

$$\tilde{g}(\beta, \delta, \tilde{y}, \tilde{b_\theta}) = \|((\frac{\sigma_d}{\sigma})^2 R + \tilde{\Lambda})^{-T/2}[\tilde{y} - \tilde{f}(\beta, \tilde{b^\theta}) - F_\delta \beta_\delta]\|^2 + \sum_{i=1}^{M} \|\Delta b_i^\theta\|^2$$

2. The Laplacian approximation to the new likelihood becomes

$$\begin{aligned} l(\beta, \Delta, \tilde{\phi}, \tilde{\rho}, \nu) &= -\tfrac{N}{2}[\log(2\pi\sigma^2)] + M \log|\Delta| - \tfrac{1}{2}\log|\tilde{G}(\beta, \Delta, \tilde{y})| \\ &\quad -\tfrac{1}{2}\sigma^{-2}\tilde{g}(\beta, \delta, \tilde{y}, \hat{\tilde{b}}_{\theta}) - \tfrac{1}{2}\log|(\tfrac{\sigma_d}{\sigma})^2 R + \tilde{\Lambda}| \end{aligned}$$

where

$$\tilde{G}(\beta, \Delta, \tilde{y}) = \frac{\partial \tilde{f}(\beta, \tilde{b^{\theta}})}{\partial \tilde{b}_i^{\theta}}\bigg|_{\hat{\tilde{b}}^{\theta}} ((\frac{\sigma_d}{\sigma})^2 R + \tilde{\Lambda})^{-1} \frac{\partial \tilde{f}(\beta, \tilde{b^{\theta}})}{\partial \tilde{b}_i^{\theta T}}\bigg|_{\hat{\tilde{b}}^{\theta}} + Diag(\Delta^T\Delta, M)$$

and $Diag(\Delta^T\Delta, M) = \begin{pmatrix} \Delta^T\Delta & 0 & \cdots & 0 \\ 0 & \Delta^T\Delta & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \Delta^T\Delta \end{pmatrix}_{M \times M}$

In the first step, we get the conditional modes of the random effects $\tilde{b_\theta}$ with β, Δ, $\tilde{\phi}$, $\tilde{\rho}$ and ν fixed. In the second step, after maximizing the approximated likelihood, we can get the estimates of β, Δ, $\tilde{\phi}$, $\tilde{\rho}$ and ν. Then these two steps are repeated until convergence.

3.4.3 Model Formulations

After introducing the technical implementation of the nonlinear mixed effects model, we can consider different types of model formulations which can all be fit by the preceding estimation procedures. Different model formulations can help us explain different types of uncertainties presented in the physical data. Through examining and comparing results of different model formulations, we could understand better the underlining computer model. The following model formulations are studied in this work.

Model 1:

$$y^F(x_i, t) = y^M(x_i, t, \theta_i) + \epsilon(t), \quad i = 1, \ldots, M \tag{3.8}$$

Here we make no change to the original computer model, so the calibrated distribution of θ will not depend on other input variables or temperatures. Also, there is no bias term, so the original computer model is expected to explain all uncertainties in the physical data.

Model 2:

$$y^F(x_i, t) = y^M(x_i, t, \theta_i(x_i, t)) + \epsilon(t), \quad i = 1, \ldots, M \tag{3.9}$$

Here we assume that the mean of the random distribution of θ will depend on design parameter x and time t. This assumption is reasonable since one of the calibration parameters, κ, is highly correlated with temperature. However, it is impossible to incorporate this temperature-dependent relationship directly into the mathematical model without significantly modifying it. McFarland (2008) considers a simple approach in which a 'representative' temperature value, T, is determined for each configuration of the system for which calibration data are available. Xiong et al. (2009) consider modeling T as a function of position x_l to accommodate the temperature dependence. There are mainly two problems with these two approaches. The first is that they ignore the time varying nature of calibration parameters. Since temperature is increasing over time, temperature-dependent calibration parameters should have similar behaviors. The second is that since they don't model calibration parameters as functions of design parameters, it will not provide much information at new untried settings. Therefore, we consider an alternative approach in which the mean of the calibration parameter, specifically μ_κ, is modeled as a function of design parameter x and time t. Again, there is no bias term, so the original computer model with design-specific time dependent calibration parameters is expected to explain all uncertainties in the physical data.

Model 3:

$$y^F(x_i, t) = y^M(x_i, t, \theta_i) + \delta(x_i, t) + \epsilon(t), \quad i = 1, \ldots, M \tag{3.10}$$

Like Model 1, the calibration parameters are assumed independent of other input variables or temperature. However, a bias term is added here to capture possible systematic uncertainty generated by the original computer model. This formulation is based on the Bayesian calibration framework developed by Kennedy and O'Hagan (2001). The bias term can be assumed to be either a linear function of other input variables or a Gaussian process function.

Model 4:

$$y^F(x_i, t) = y^M(x_i, t, \theta_i(x_i, t)) + \delta(x_i, t) + \epsilon(t), \quad i = 1, \ldots, M \tag{3.11}$$

This is a combination of Model 2 and Model 3. To some extent, it can be thought as a full model like in traditional regression. It is not necessarily good to be as comprehensive as model 4, however it is our interest to compare results and performances of different model formulations.

3.5 Summary of Thermal Problem Results

The nonlinear mixed effects models without bias function or with linear bias function are implemented in R, a freely available statistical software package. The library we are using calls "nlme" which is written by Pinheiro and Bates. We use Matlab 7.6 to implement the Gaussian process based models.

Before we get to the results, the details of different model formulations are summarized in Table 3.2.

Table 3.2: Model components and assumptions

	Function Form	Bias Form	Fixed Effects	Error Structure
Model 1	$f(x, \theta)$	No	$\theta_0 = (\mu_\kappa, \mu_{\rho Cp})$	AR1$(h(t) = \rho^t)$
Model 2	$f(x, \theta(x, t))$	No	θ_0, β_θ	AR1
Model 3	$f(x, \theta)$	$\delta(x, t) = f_\delta^T(x, t)\beta_\delta$	θ_0	AR1
Model 4	$f(x, \theta(x, t))$	$\delta(x, t) = f_\delta^T(x, t)\beta_\delta$	θ_0, β_θ	AR1
Model 5	$f(x, \theta)$	$\delta(x, \theta, t) \sim N(0, \sigma^2 R)$	θ_0	No
Model 6	$f(x, \theta(x, t))$	$\delta(x, \theta, t) \sim N(0, \sigma^2 R))$	θ_0	No
Model 7	$f(x, \theta)$	$\delta(x, \theta, t) \sim N(f_\delta^T(x, t)\beta_\delta, \sigma^2 R)$	θ_0	No
Model 8	$f(x, \theta(x, t))$	$\delta(x, \theta, t) \sim N(f_\delta^T(x, t)\beta_\delta, \sigma^2 R)$	θ_0	No

Among those models, Model 5 and 6 are extensions of Model 3 and Model 7 and 8 are extensions of Model 4. And for hyperparamters in the Gaussian process model, we assume that $\rho_x = 2$ and $\rho_\theta = 2$ which correspond to the Gaussian correlation function.

3.5.1 Parameter Estimation & Calibration Results

The following table (Table 3.3) presents the estimation results from different model formulations. Since the measurements for one experimental run (group) are recorded consecutively, the within-group errors of Model 1-4 are assumed to be autocorrelated over time. In Model 5-8, we assume that there is no random errors ($\epsilon = 0$). This assumption is due to the fact that the measurements are claimed to be highly accurate which suggests that σ^2_ϵ is relatively small compared to σ^2_δ. This assumption also makes possible to profile the loglikelihood on σ^2_δ instead of estimating it together with other paramters.

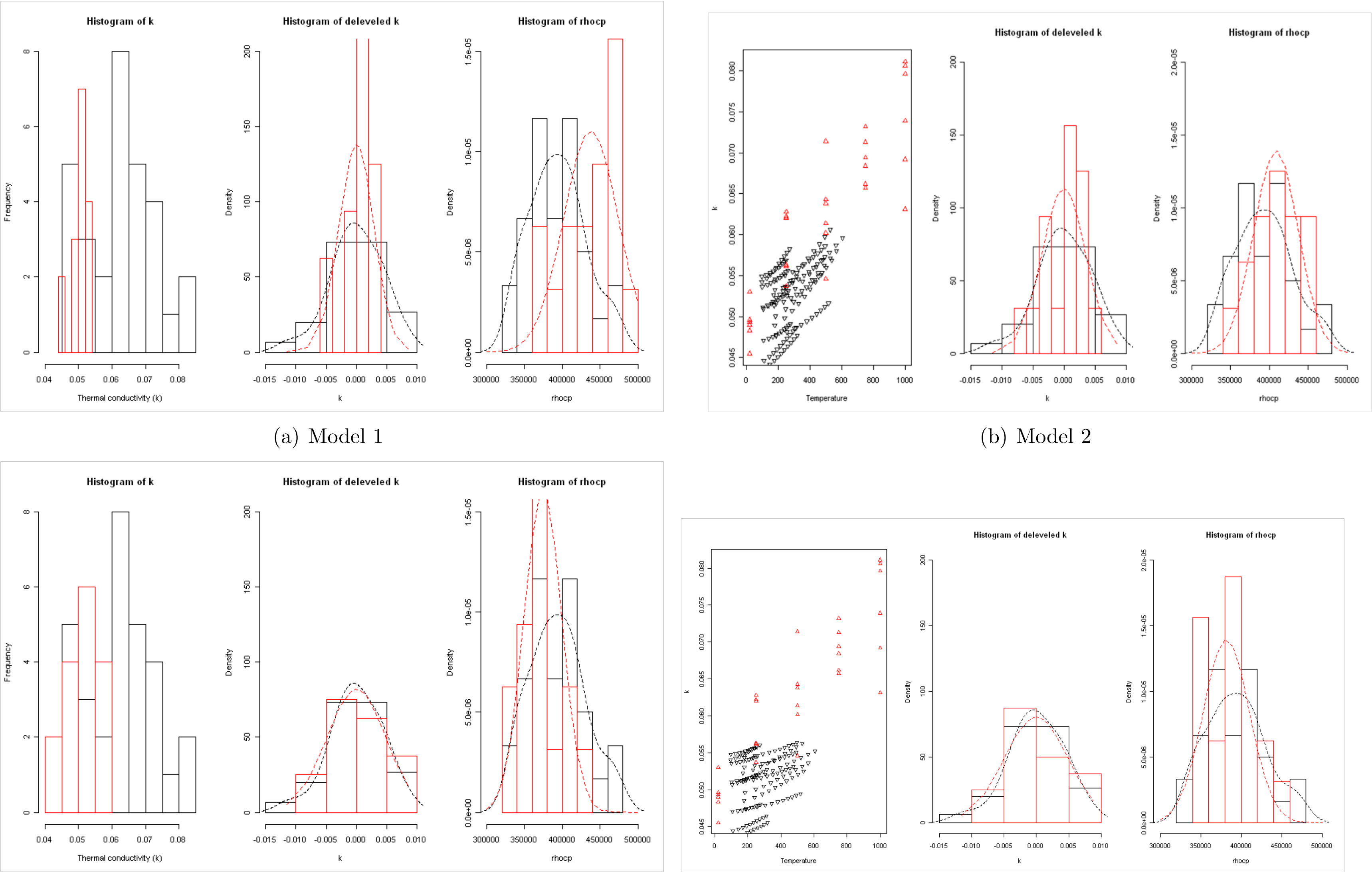
Table 3.3: Estimation results from Model 1-8

	κ_0	σ_{k_0}	β_κ	$\rho C p_0$(1E5)	$\sigma_{\rho C p_0}$	ρ	$\sigma_\epsilon/\sigma_\delta$
Model 1	0.0502	0.00289	0	4.38	0.361	0.983	3.636
Model 2	0.0502	0.00354	5.60E-06	4.08	0.287	0.996	2
Model 3 (Linear)	0.0512	0.00488	0	3.74	0.247	0.993	0.848
Model 4 (Linear)	0.0522	0.00496	2.80E-06	3.82	0.285	0.992	0.71
Model 5 (GP1)	0.0493	0.00276	0	4.45	0.422	no	2.4514
Model 6 (GP1)	0.0504	0.00239	5.20E-06	4.11	0.257	no	2.7431
Model 7 (GP2)	0.0503	0.0028	0	3.99	0.284	no	1.7935
Model 8 (GP2)	0.0495	0.008	4.27E-06	4.29	0.879	no	0.3229

Fig 3.5 and 3.6 show the calibration results compared to material characterization data based on Model 1-4 and Model 5-8, respectively. The goal here is to match the calibrated parameter distributions to measured material data as close as possible. The first graph in each subplots compares histograms of observed thermal conductivities (κ) and estimated random effects $\hat{\kappa}_i$'s (=$\hat{\kappa_0}$ + $\hat{b^\kappa}_i$) for Model 1, 3, 5 and 7 or scatterplots of observed thermal conductivities (κ) and estimated $\hat{\kappa}_i$'s versus temperatures for Model 2, 4, 6 and 8. The middle graph in each subplots compares the histograms of deleveled thermal conductivity (κ) and estimated random effects $\hat{b^\kappa}_i$'s.

The 'delevel' means that the mean value of κ's of the same temperature is subtracted from individual observations accordingly. This allows us to compare the true variation of observed thermal conductivities. The third graph in each subplots compares the histograms of observed heat capacities, ρCp, and estimated $\rho \hat{C}p$'s.

Apparently, the estimated means of ρCp in Figure 3.5(a) and 3.6(a) deviate significantly from the observed mean of ρCp due to the fact that we don't consider the relationship between κ and temperature and no bias function is used to correct the original computer model. The estimated variation of calibration parameters in Figure 3.6(d) are much larger than the true variation which suggests that the estimation results may not be very reliable.



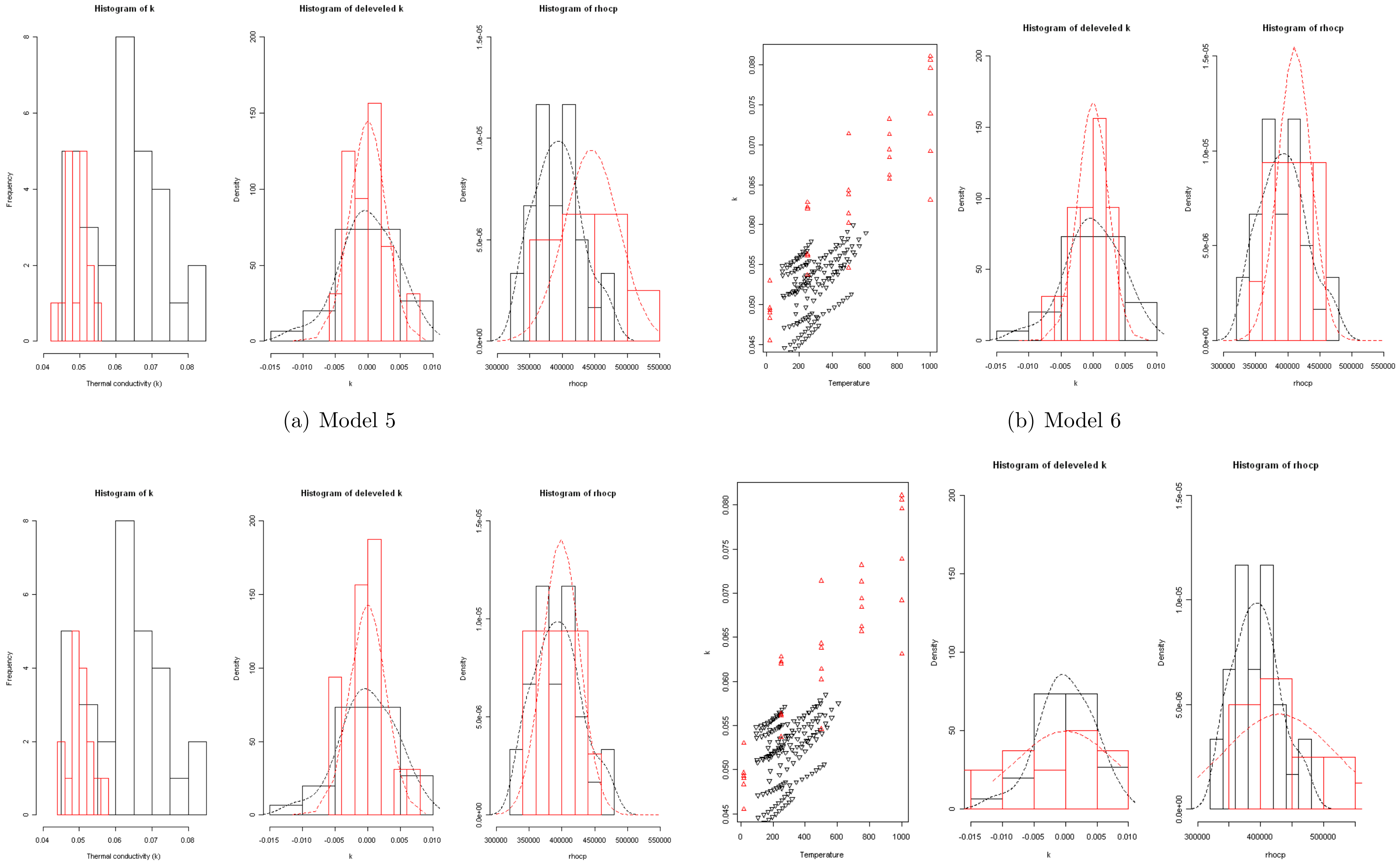
(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

Figure 3.5: Calibrated distribution compared to material characterization data (Model 1-4)



(a) Model 5

(b) Model 6

(c) Model 7

(d) Model 8

Figure 3.6: Calibrated distributions compared to material characterization data (Model 5-8)

3.5.2 Prediction and Validation

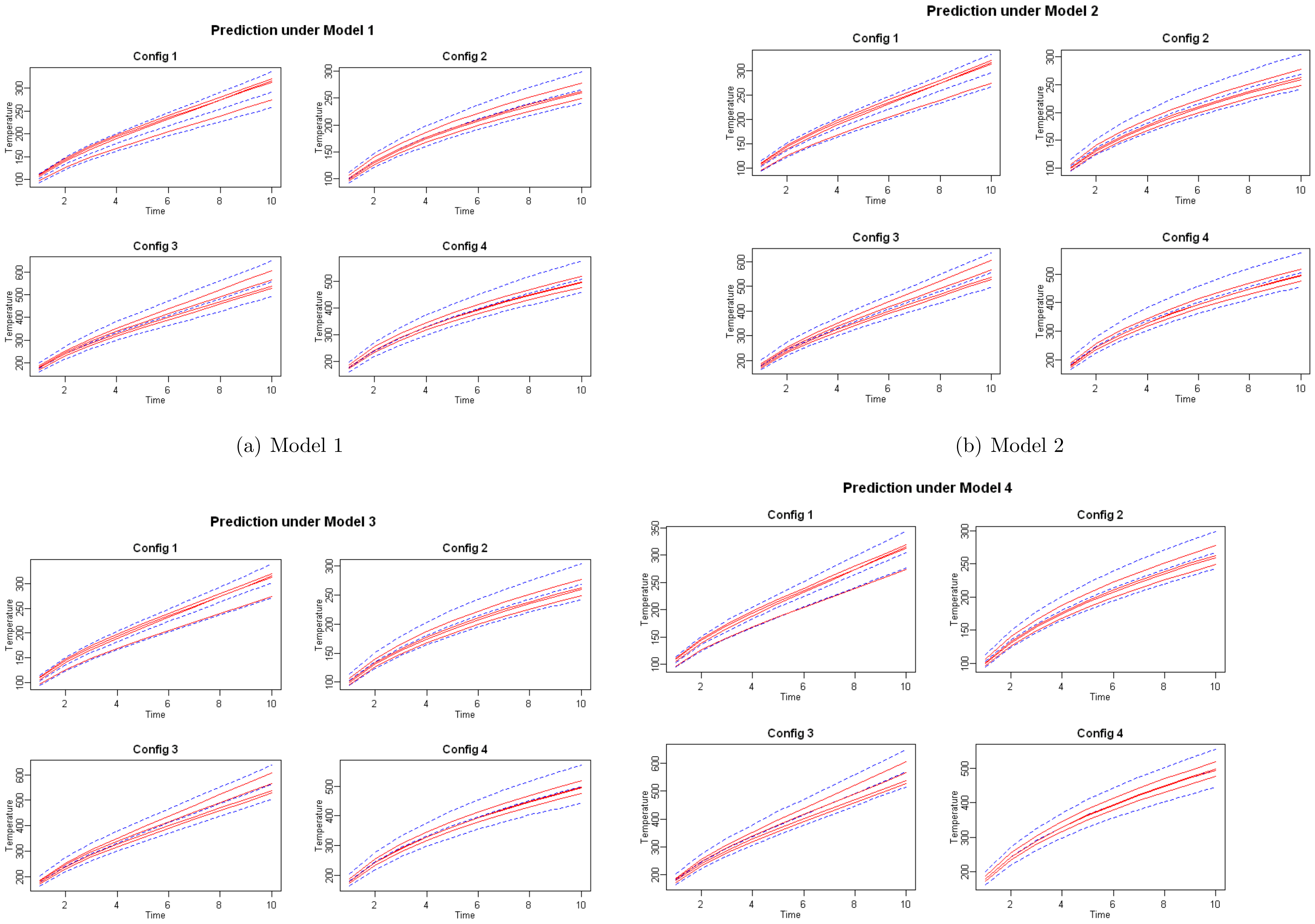
3.5.2.1 Prediction

The prediction under Gaussian process bias function is a natural extension of standard kriging prediction. Following (2.3), given new x^*, t^* and θ^*,

$$\begin{aligned} \hat{\delta}(x^*, t^*, \theta^*) &= E(\delta(x^*, t^*, \theta^*)|y) && (3.12) \\ &= f_\delta(x^*, t^*, \theta^*)\hat{\beta}_\delta + rR^{-1}(y^F - y^M(x, t, \hat{\theta}) - F_\delta\hat{\beta}_\delta) && (3.13) \\ \hat{y}(x^*, t^*, \theta^*) &= f(x^*, t^*, , \theta^*) + \hat{\delta}(x^*, t^*, \theta^*) && (3.14) \end{aligned}$$

where $r = (c((x^*, t^*, \theta^*), (x_1, t_1, \theta_1)), \ldots, c((x^*, t^*, \theta^*), (x_N, t_N, \theta_N)))$.

In reality, we only know x^* and t^* at which we want to predict. Therefore, we have to simulate θ^* from the calibrated distribution. The prediction at (x^*, t^*) will be a collection of points and the prediction at x^* will be a collection of functional curves. Figure 3.7 and 3.8 show predictions at configurations 1-4 for Model 1-4 and 5-7 respectively. The red solid lines represent the physical observations at those configurations. The blue dash lines represent the 99%, median and 1% quantile curves of predictions. The predictions of Model 8 are not shown here since the estimated correlation matrix is not stable which gives unreliable results. It will be our future work to investigate the possible causes of such instability and improve the estimation procedures for Model 8.



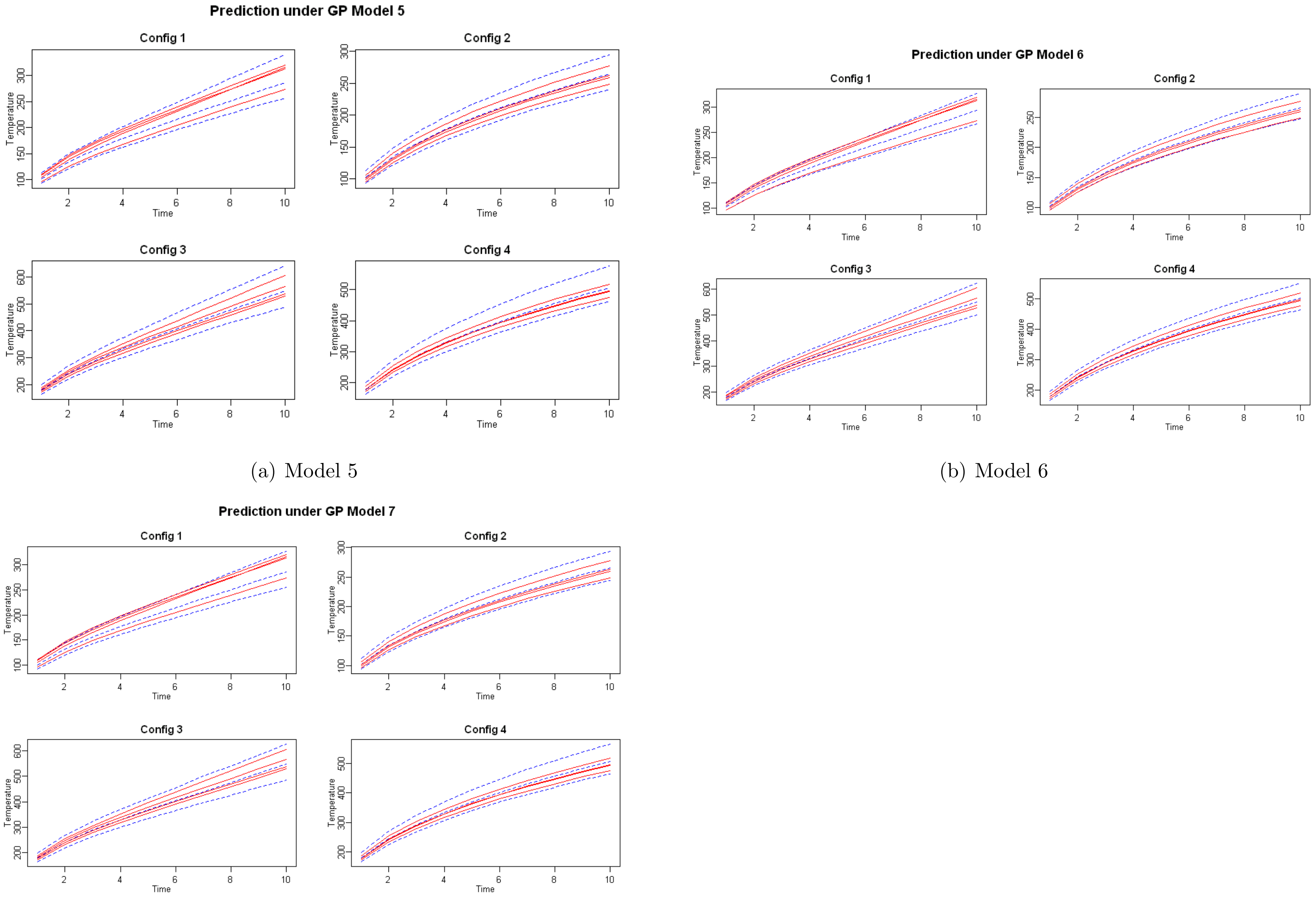
(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

Figure 3.7: Prediction distributions under 4 different configurations (Model 1-4)



(a) Model 5

(b) Model 6

(c) Model 7

Figure 3.8: Prediction distributions under 3 different configurations (Model 5-7)

3.5.2.2 Validation

It is very difficult to judge the performance of different models based solely on graphical comparisons, therefore a validation metric is called on to provide a quantitative measure of the performance. In this paper, we adopt a modified version of the u-pooling method recently developed by Ferson et al. (2008) for model validation. As we have introduced in Section 2.2, a nice feature of the u-pooling method is that it allows to integrate or pool all available physical experiments over a validation domain at different input settings x into a single aggregate metric. First, a value u_i is obtained for each experiment by calculating the CDF at y_i^e, i.e., $u_i = F_{x_i}(y_i^e)$ $(i = 1, \ldots, N)$, where y_i^e represents a physical observation at the experimental site x_i. $F_{x_i}(y_i^e)$ represents the corresponding CDF generated by the statistical model at x_i. According to Ferson et al., if each physical observation y_i^e hypothetically comes from the same 'mother' distribution $F_{x_i}(\cdot)$,all u_i's are expecte toconstitute a standard uniform distribution on $[0, 1]$. An illustration of the u-pooling method with three experimental sites is given in Fig 3.9. By comparing the empirical distribution of u_i to that of the standard uniform distribution, the area difference (depicted as the shaded region in Fig 3.9) can be used to quantify the mismatch between the dispersion of physical experiments and the distributions of model output.

Following their idea using u_i to pool information together, we consider the non-parametric Cram$\acute{e}$r-von-Mises test statistics which is defined as

$$T = \frac{1}{12n} + \sum_{i=1}^{n} \left[\frac{2i-1}{2n} - u_i' \right]^2 . \tag{3.15}$$

u_i''s are rank statistics of u_i's.

The metric under the original model is calculated by using simulations directly sampled from fitted distribution of material charactarization data. It should be noticed that in terms of prediction, the Gaussian process based models don't perform well compared to nonlinear mixed effects models with linear bias functions. There

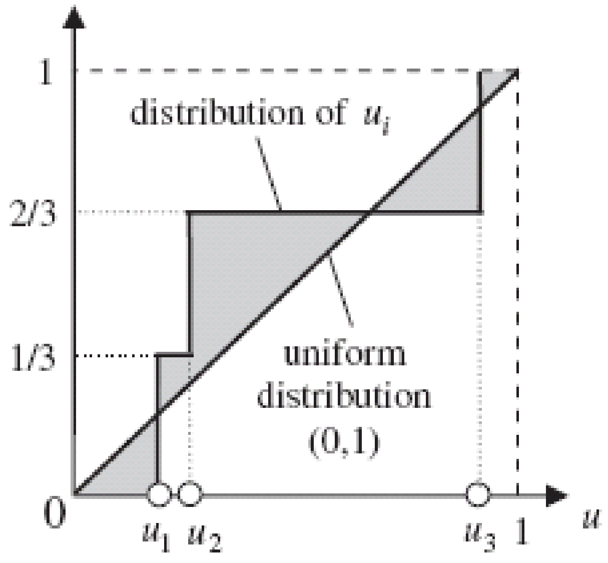


Figure 3.9: Illustration of the u-pooling method.

are two possible reasons: the Gaussian process models (Model 5-7) tend to underestimate the variance of calibration parameters and there is an strong interaction between random effects and the Gaussian process bias funtion which leads to unstable estimations. In general, all statistical models performs much better than the original computer model without careful calibration. And more likely Model 4 should be considered for future inference.

Table 3.4: Validation metrics from different models

Model	Original	1	2	3	4	5	6	7
Metrics	0.303	0.076	0.085	0.069	0.065	0.078	0.159	0.125

3.6 Discussion

Different model formulations and combinations with Gaussian process bias functions can help us much better understand uncertainties involved in this application. Figure 3.5 and 3.6 provide a way to evaluate performances of different approaches by comparing calibrated distributions to independently measured data. Clearly, adding more structures to calibration parameters help getting more desired results. Also, a flexible Gaussian process assumption on bias function tends to give better results than much restricted linear functions.

From the prediction point of view, it is not very clear which formulation or type of bias function outperforms others. One possible reason is that computer model itself is already capable enough on the design domain, which is far away from the testing area. To improve prediction performance, it is much more helpful to add design points which are closer to the testing area. The other alternative is that we may want to generate some data from computer models of higher fidelity.

The proposed approach is based on the nonlinear mixed effects model which requires a explicit function form of computer model. However, it is not limited to this kind of computer experiments. When the computer model becomes expensive to run, we can build a surrogate model first and then treat it as the function in the NLME model.

3.7 Future Works

The previous results show that there are significant interaction effects between two calibration parameters and the Gaussian process bias function. It will be necessary to conduct various simulation studies to understand how the interaction effects have influence on model results.

The approximation of the loglikelihood function used in the current research is not accurate enough when the bias function is assumed to follow a Gaussian process. To improve the credibility of the proposed modeling approach, either a new approximation or a EM type method based on the exact loglikelihood should be considered in future research.

The MLE implementation of the proposed approach is easy to understand and carry out for computer model users or engineers. However, it is difficult to consider and understand the full uncertainty of all parameters and the model itself. This limitation calls on combining Bayesian analysis with the proposed approach. Nagy et al. (2008) consider an new approach to combine MLE and Bayesian inference. Following their idea, we could develop a similar approach for our proposed model.

CHAPTER IV

REGRESSION MODELING FOR COMPUTER MODEL VALIDATION WITH FUNCTIONAL RESPONSES

4.1 Introduction

In recent years, functional data have arised in many engineering applications with dynamic computer models, e.g. vehicle crashworthiness test (Bayarri et al. (2005)), dynamic stress analysis (Bayarri et al. (2006)), and response histories study (Schwer, L.E. (2007)). Therefore, it is very important to extend the existing additive bias function approach for functional responses that are not only functions of physical attributes but also functions of time or space. The major challenges for dealing with functional outputs are complexity and high-dimensionality, which make it infeasible to either apply the traditional frequentist approach or the computation-intensive Bayesian approach. Bayarri et al. (2005) considered a Gaussian process approach that considers time as an additional model input. However, their approach will become infeasible if there are a large amount of time points (e.g, 1000), which makes the algorithm extremely inefficient. Bayarri et.al. (2006) considered an alternative approach which utilizes wavelet decomposition for highly irregular functional data. However, owing to its computational complexity, it is difficult to implement the method.

A lot of literatures exist on functional data analysis based on longitudinal data in biological science application, in which data are characterized by relatively few measurements per individual with large experimental errors. The individual's progress or curve is often approximated by a simple parametric form. However, as we have more accurate and well-controlled measurements for each individual or unit, it may not be reasonable to make strong parametric assumptions. Recently, functional data analysis has become more popular due to the availability of large amount of accurate measurements. Faraway (1997) had proposed a functional regression analysis approach for ergonomically correct design of equipment. Nair et.al. (2002) applied the similar approach to analyze some robust design studies. Ramsay et al. (2002) gave a comprehensive review of functional regression models.

Oberkampf and Barone (2004) pointed out that regression functions are commonly used to approximate the bias function between computer outputs and physical experiments when the input variables are all physical parameters. In this paper, we propose a piecewise regression approach for computer model validation with functional outputs. Specifically, the first step is to build a functional regression model for fitting functional data with multiple input variables. Secondly, we will use this model to fit the bias function between physical and computer outputs. The proposed approach is expected to be efficient for a smooth functional response with a large amount of time points. As illustrated in this paper, the approach can be easily implemented and combined with validation metrics for assessing the accuracy of a computer model.

This chapter is organized as follows. Section 4.2 presents the formulation of the functional regression model and methods of analysis for the purpose of validation. A real industrial application is described in Section 4.3. In Section 4.4, we will introduce a new approach based on Gaussian process and illustrate it using the same application. We conclude our work in Section 4.5, followed by some future works in Section 4.6.

4.2 Models and Methods of Analysis

4.2.1 Functional Responses

Unlike single-output systems, the responses of multiple-output or real time system is of functional forms, i.e. $y_i(t)$ for run i under a specific model input condition x. In practice, however, we only observe $y_i(t)$ at discrete time points, i.e. $y_i(t_{ij})$,

$j = 1, \ldots, m_i$. The number of measurements, m_i, will not always be the same for every run under setting x. This is true because the time interval for shift events varies for every run. In that case, under a specific criterion, the experiment may be stopped at different time intervals for every run. Our attempt is to reconstruct the curves $y_i(t)$ based on functional data as well as to predict the time interval for shift events.

Faraway (1997) mentioned that there are two kinds of variability here, variability specific to the particular observations, $y_i(t_{ij})$, and variability related to the whole functional curve, $y_i(t)$. It is not easy to separate these two types of variability. Smooth estimation has been introduced to eliminate (or at least reduce) the variability of the first kind (Ramsay et.al. (2002)).

One approach is to smooth each y_i individually without reference to the particular model being fit. Various parametric or nonparametric regression model could be considered here and the smoothing parameters could be selected automatically in the presence of correlated errors. We are particularly interested in using basis functions to represent functional curves. A basis function system is a set of known functions ϕ_k that are mathematically independent of each other and have the property that any function can be approximated arbitrarily well by choosing a weighted sum or *linear combination* of a sufficiently large number K of these functions. Spline functions are the popular choice of approximation system for non-periodic function data. In this paper, we focus on spline functions with order no more than two.

4.2.2 Functional Regression Model

Suppose that the functional responses, $y_i(t)$, arise from the model $y = X\beta + \epsilon$, where β is a vector of functions $(\beta_1(t), \ldots, \beta_p(t))^T$ and X is the $n \times p$ design matrix formed from the p-vector valued covariates x_i, $i = 1, \ldots, n$. Moreover, y is a vector of response functions $(y_1(t), \ldots, y_n(t))^T$ and ϵ is a vector of error functions $(\epsilon_1(t), \ldots, \epsilon_n(t))^T$. Each $\epsilon_i(t)$ is an independent realization of a stochastic process with mean 0 and variance σ^2.

To create a functional model that best matches with the data, we choose $\hat{\beta}$ to minimize $\sum_{i=1}^{n} \|y_i - x_i^T \beta\|^2$. The minimum can be achieved by minimizing each t_i individually, which turns out to be a pointwise regression. In fact, Faraway (1997) used this approach to analyze the body motion data in his study.

In the pointwise approach, we tend to ignore the parametric relationship between y and t. However, this approach is questionable if the relationship between y and t is simple and smooth. Therefore, our model becomes

$$y(t) = \sum_{i=0} X\beta_i I\{t_i \le t < t_{i+1}\} + \sum_{i=0} X\xi_i t I\{t_i \le t < t_{i+1}\} + \epsilon \tag{4.1}$$

After assigning all t_i's, fitting such a model will give us a piecewise linear regression.

More specifically, we have k+1 knots $t_0, t_2, \ldots, t_k$ where $t_0 = 0$ and t_k is associated with the last observation over the time interval for each run under a specific model input condition.

In reality, very often we observe that the functional responses tend to change smoothly over time which will be illustrated in the example in Section 3. To achieve smoothness in the piecewise linear model, we choose k+1 basis functions (order two spline functions):

$$h_1(t) = 1, h_2(t) = t, h_3(t) = (t - t_1)^+, \ldots, h_{k+1} = (t - t_{k-1})^+$$

$$y(t) = \sum_{i=1}^{k+1} X\beta_i h_i(t) + \epsilon, \tag{4.2}$$

where β_i is a $p \times 1$ vector and X, design matrix, is a $n \times p$ matrix. It is noted that the term $\beta(t)$ in the general model $y(t) = X\beta(t) + \epsilon$ becomes $\sum_{i=1}^{k+1} \beta_i h_i(t)$. Such formulation enables us to model the data globally instead of pointwisely.

4.2.3 Modeling for Interval Shifting Events

As mentioned in Section 4.2.1, for some engineering problems, the interval of interest (the shift interval) can vary for different inputs x under different runs. The conventional approach models the interval as a function of only input variables. However, the approach becomes questionable if the shift intervals and functional responses are strongly correlated with each other. In that case, the conventional approach may not be able to provide an accurate prediction due to ignoring this high correlation. Therefore, we propose a two-stage approach which allows prediction of both the shift time and the response at the shift time.

1. Rescale all t_{ij}, $j = 1, \ldots, m_i$ to be within $[0, 1]$ and denote the rescaled time as t'_{ij}
2. Build a functional regression model for response profile under scaled time: $y(t') = f(x, t')$

   Predict the response at the shift time at a new setting: $y_e = f(x, 1)$
3. Build a regression model for the shift time: $T_s = f_T(x, y_e)$

   Predict the shift time at the new setting using the predicted value of y_e obtained in Step 2.

This two-stage approach (Step 2 and 3) takes into account the relationship between the shift time and the functional responses, which is expected to significantly improve the accuracy of a prediction model.

4.2.4 Computer Model Validation for Functional Model

There are few literatures in validation of computer model with functional outputs. For single-output systems, the existing model validation approach is to validating a computer model by directly comparing the results from both computer and physical experiments.

An alternative design validation approach, proposed by Chen et al. (2007), emphasized on enhancing the predictive capability of a computer model for the purpose of design decision making. Recent methods under existing model validation approach can be divided into two categories, namely classical frequentist approach and Bayesian approach. Under frequentist's approach, linear or nonlinear regression models are fitted for bias function using physical and computer outputs and then a validation metric could be built according to the fitted bias function. Under Bayesian's approach, appropriate priors of certain parameters are assumed and the posterior distribution of the model bias serves as a basis for the quantitative comparison of computer outputs and physical observations. For the design validation approach, a bias-corrected computer model is first built by characterizing the bias function between the computer model and physical experiments. Together with the uncertainty quantification of a bias function, the approach provides a confidence assessment of a design alternative being superior to other alternatives in terms of their design objective function values.

It is noted that the bias function is widely used in computer model validation literature to provide a basis for comparing computer and physical outputs (Chen et.al. (2006)). It can also capture the potential model or method error, which sometimes cannot be compensated for by other means. The basic bias-correction model for a single output is given as follows.

$$y^e = y^m + \delta + \epsilon, \tag{4.3}$$

$$y^r = y^m + \delta, \tag{4.4}$$

where y^r is the true response, y^m is the computer model output , δ stands for the discrepancy between reality and a computer model (also called the bias function) and ϵ is the experimental error. This model has been used in a Bayesian approach for computer model validation (Kennedy, M.C. and O'Hagan, A. (2001) and Wang et.al. (2007)). In that approach, two Gaussian process models are assumed for y^m and δ

and a Bayesian approach is used to provide prediction and uncertainty quantification of both $\hat{\delta}$ and $\hat{y}^r$.

In the functional regression model, a similar bias-correction model can be defined at each time points, i.e.,

$$y^r(t) = y^m(t) + \delta(t). \tag{4.5}$$

Similarly, another functional regression model is fitted to the bias profiles between computer outputs and physical observations, i.e.,

$$\delta(t) = \sum_{i=1}^{k+1} X\beta_i h_i^{\delta}(t) \tag{4.6}$$

where $h_i^{\delta}(t)$'s are defined similarly as $h_i(t)$'s in Section 4.2.2.

4.3 An Industrial Application

Computer simulation is a powerful tool for investigating complex transmission systems. This can lead to shorter product design cycles, reduce development cost, and allow engineers to explore many options early in the design phase. Simulating the transient characteristics of an automatic transmission is, however, complicated because many factors affect the shift quality during gear changes. In our problem, a computer model provided by industry is used to simulate acceleration performance and gear shift events of motor engine under various conditions. The complete model consists of two parts: a drive-line model which produces acceleration values and a transmission model which produces torque values and decide the time of shift event. The flow of computer engine system modeling is illustrated in Figure 4.1.

4.3.1 Experiment Data

Physical experiments and computer experiments were carried out as follows. For physical experiments, there are four input parameters: Tire coefficient (low or high), Drag coefficient (low or high), Road grade (slope, 4 choices), and Throttle position (4 choices). A $4^2 2^{2-1}$ fractional factorial design (32 combinations) is chosen. At

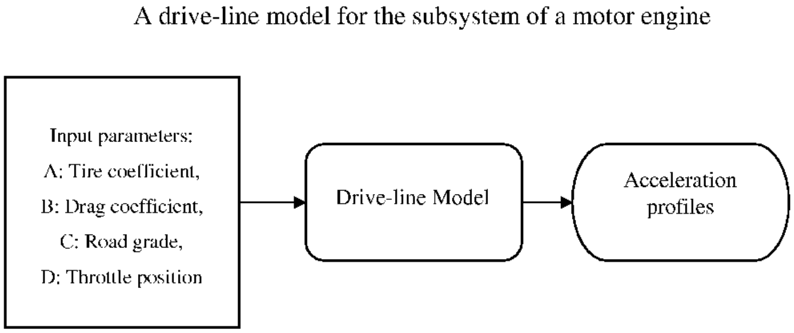


Figure 4.1: A complex transmission system for computer model

each combination, the vehicle was driven twice, i.e., two replicates for each design combination. The acceleration and torque values were recorded every 0.002 seconds until the first gear shift occurred. Finally, the shift times were recorded for each experiment. For computer experiments, there are several submodels in the mathematical model which correspond to different subsystems in physical engine system. In this analysis, we isolate the drive-line model (more like a transmission system) from the entire system. For this drive-line model, it used torque values from physical experiments as another input variable to generate acceleration profiles for each input combination. Our focus in to validate the isolated drive-line model, so the torque values are not considered as an input variable. Finally, functional acceleration values with shift times were collected from both physical and computer experiments for all 64 (32 × 2) runs. The experiment description is summarized as follows.

4.3.1.1 Acceleration Model Parameter Definition

Responses	Acceleration Profile $y_i(t)$, Shifting Time t_i
Input Variables	A (Tire coefficient (low or high)), B (Drag coefficient (low or high)), C (Road grade (slope; 4 choices)), D (Throttle position (4 choices)
Dynamic Variable	Torque Profile $q(t)$

4.3.1.2 Experimental Design

A $4^2 2^{2-1}$ fractional factorial design is planned as follows.

- A 2^5 full factorial design table is created.
- Denote each column as 1, 2, 3, 4, 5.
- Replace two columns 1, 2 by a four-level column C and two columns 3, 4 by a four-level column D. Assign 5 to a column B and finally assign 12345 to a column A.

The defining word of this fractional factorial design is $A = BC_3D_3$, where $C_3 = C_1C_2$ and $D_3 = D_1D_2$.

4.3.1.3 Objectives

We set three study objectives:

1. Prediction of Acceleration Profiles based on Functional Data
2. Prediction of Ending Acceleration and Shifting Time
3. Prediction of Bias Function for Validation of Functional Computer Model

4.3.2 Rescaling and Data Preprocessing

As explained in Section 4.2.3, to effectively predict both functional responses and the shift time, we rescale $[0, t_{shift}]$ to $[0, 1]$ by dividing each t by t_{shift}. Thus, t_{scaled} does not represent time, merely the proportion of the acceleration between the start and the first shift. For a given combination, $y(t)$ is observed on an equally spaced grid of points, but the number of such points varies from experiment to experiment.

A plot of physical observation data after this rescaling is shown in Figure 4.2.

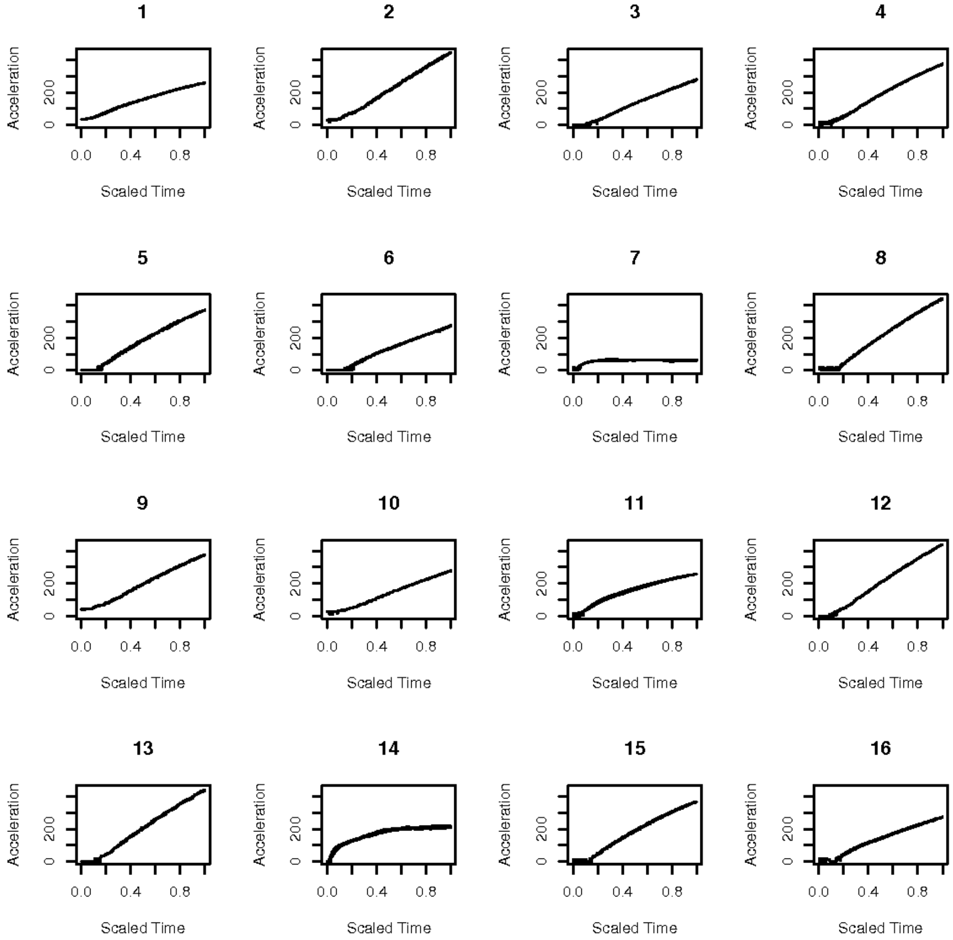


Figure 4.2: Field Observation (16 Combinations, each plot consists of 2 replicates)

One issue for this dataset is that there are several aberrant combinations observed from physical experiments. For this specific problem, we have found that when D=0 and $C > 0.5$, the acceleration profile and shift time are very different from other settings (See Figures 4.3 and 4.4). The dashed line in Figure 4.4 clearly separates the regular combinations from the outlier combinations.

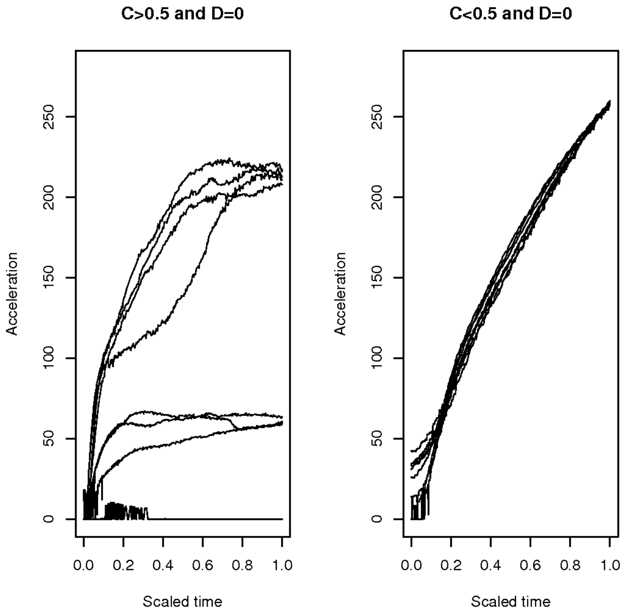


Figure 4.3: Left panel: Outlier combinations; Right panel: Regular combinations

After initial analysis, we concluded that the outliers need to be eliminated first to ensure accuracy of our analysis. Therefore, the eight problematic points (four combinations, each with two replicates) were removed from our analysis; hence the final data set consists of 56 runs and 28 combinations.

Finally, the original data set contains too many time points that can be handled in limited computer resource environment. To overcome this difficulty, without sacrificing the prediction accuracy, we used a subset of samples generated from the entire dataset by including 250 data points from the rescaled time period 0-0.2 and 250 data points from the rescaled time period 0.2-1.

4.3.3 Prediction of Acceleration Profiles

In this section, we illustrate how the proposed approach can be used to model and predict acceleration profiles. We only consider physical experiment data here, since

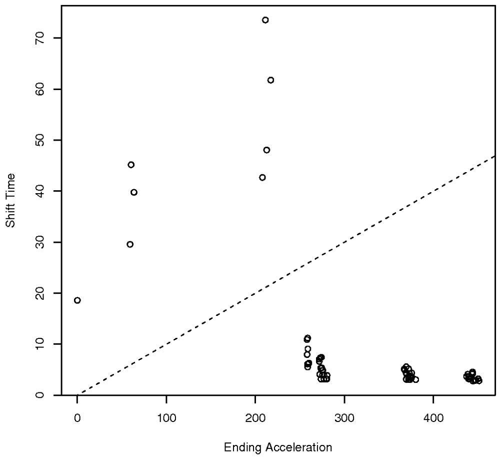


Figure 4.4: Ending Acceleration vs Shift Time

it is sufficient to demonstrate the strength of the proposed method.

For those 56 runs, excluding outliers described in Section 3.3, we randomly chose ten runs as testing (confirmation) sample and the remaining 46 runs as training sample. In the ten runs of testing sample, four of them are replicates from two different combinations. Therefore, we have 26 combinations in the training sample with 20 of them being replicated. The proposed functional regression model described in Section 2 was fitted to the training data and the fitted model is used to predict the acceleration profiles for the testing data.

Figure 4.5 shows the fitted acceleration profiles and the original acceleration profiles. Figure 4.6 shows the predicted acceleration profiles for the two combinations with replicates in the testing sample. Both the fitting and prediction look quite satisfactory based on visual inspection.

4.3.4 Prediction of Ending Acceleration and Shift Time

With the functional regression method we proposed, we can predict the ending acceleration properly and independent of the shift time based on the idea of rescaling.

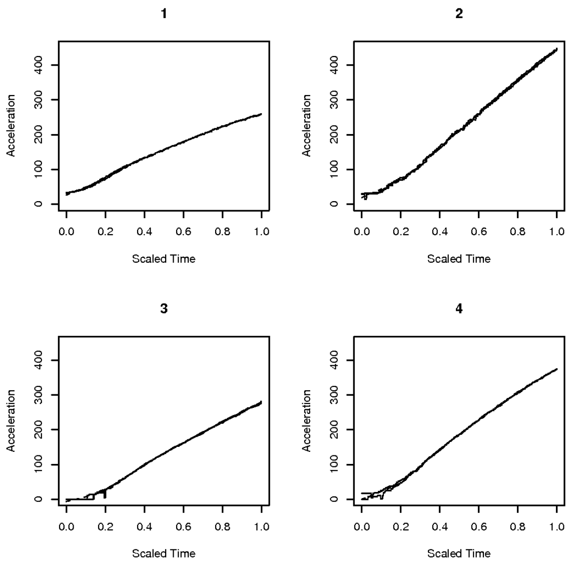


Figure 4.5: Fitted Acceleration for 4 Combinations of Inputs: Observation (solid line), Fitted (dashed line)

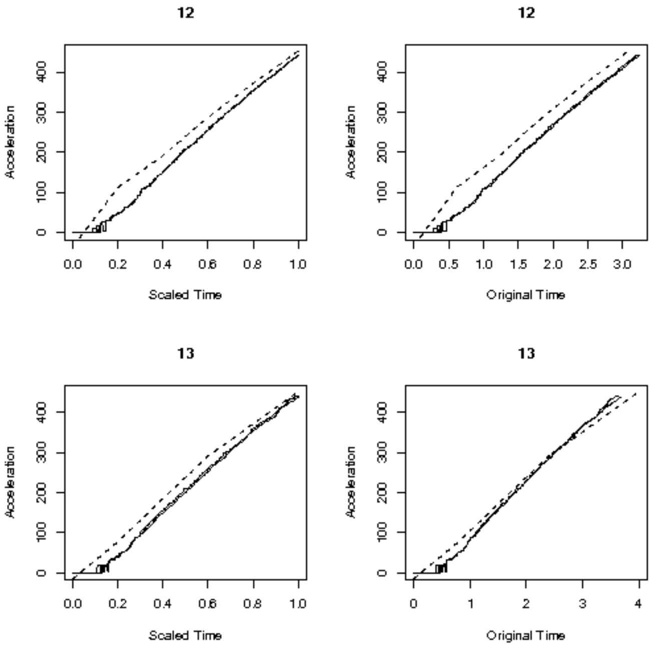


Figure 4.6: Prediction for 2 Combinations: Observation (solid line), Prediction (dashed line)

Using the approach proposed in Section 4.2.3, the model for shift time becomes

$$t_{shift} \sim f(A, B, C, D, y_{end})$$

. Following Section 4.2.3, the two stage approach for prediction of shift time and ending acceleration is listed as follows.

1. Build a functional regression model for acceleration profile under scaled time: $A(t) = f_A(A, B, C, D, t)$
   Predict ending acceleration for a new setting: $A_e = f_A(A, B, C, D, 1)$
2. Build a regression model for shift time: $T_s = f_T(A, B, C, D, A_e)$
   Predict shift time for a new setting and using predicted values of A_e

Table 4.1 compares the prediction of shift time using our proposed two-stage approach versus an approach using the regression model without ending acceleration. For comparison, we fitted a simple regression model of $T_s = f_t(A, B, C, D)$ without taking A_e as an input and used the training data to predict the testing settings.

As indicated by the results in Table 4.1, prediction without the ending acceleration can be quite poor for combinations not in the training sample. We find that the proposed procedure significantly improves the prediction of shift time for the testing settings. As shown in Figure 4.7, the plot of shift time versus ending acceleration indicates that a simple linear regression model without considering ending acceleration is inadequate.

4.3.5 Validation of Computer Model to Physical Process

As we have pointed out before, the bias function plays a very important role in computer model validation. Following the formulation in Section 4.2.4, functional regression analysis can be applied to build a functional regression model for the bias function. Afterwards, either a specific validation metric can be applied to verify the

Table 4.1: Comparison of Two models for Prediction of Shift Time

Combination	**5**	**9**	**12**	**12**	**13**
Observation	4.444	3.068	3.230	3.204	3.678
Proposed Method	4.916582	3.349589	3.075773	3.075773	3.999299
Regression w/o A_e	4.041277	2.968270	10.545518	10.545518	8.502972
Combination	**13**	**15**	**20**	**29**	**30**
Observation	3.624	4.942	3.212	4.206	5.310
Proposed Method	3.999299	5.044591	2.908936	4.912238	6.254730
Regression w/o A_e	8.502972	5.679730	3.212000	4.006397	5.217603

RMSE of Proposed Method: 0.45613553

RMSE of Regression w/o A_e: 3.940357455

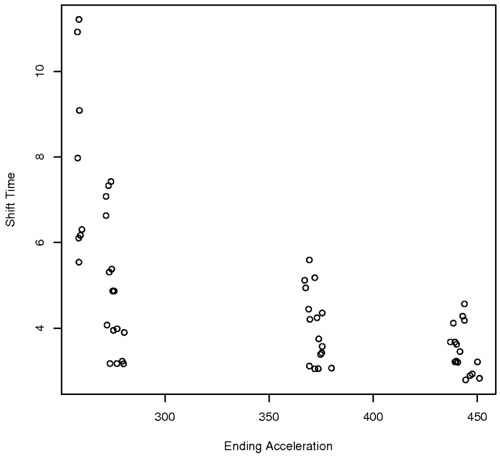


Figure 4.7: Ending Acceleration vs Shift Time (56 runs)

validity of the original computational model or the prediction of bias function can be used to update the existing computer model to achieve better accuracy.

In this particular industrial application, the bias profiles are significant over all design combinations, meaning the original computer model is not sufficiently accurate. Therefore, our objective is to see if combining computer outputs and physical observations gives more accurate predictions of physical process, or whether an updated computer model can improve the accuracy of the computer model outputs.

The procedure for model validation in this application is listed as follows.

1. We randomly divided our data into two groups: training group (for bias function fitting) and validation group.

2. A functional regression model was fitted to the training group for the bias function using equation (6).

3. We predicted the bias functions for the validation group using the fitted functional regression model.

4. We combined the computer outputs in the validation group with the prediction of bias functions using Eqn (5) to obtain an updated computer model.

5. Validation metrics were applied to the updated computer outputs against the physical outputs in the validation group.

In the last step, we need to choose a functional validation metric for our problem. Schwer, L.E. (2007) gives a clear demonstration of two existing validation metrics for comparing measured and simulated response histories: the Sprague and Geers metric and the Knowles and Gear metric. In this paper, we illustrate our example using the Sprague and Geers metric which combines magnitude and phase differences between different functional responses in comparison. The metric is also general and easier to implement.

The definition of Sprague and Geers metric is given in the Appendix. Figure 4.8 shows the prediction of the bias function and the updated computer outputs for one setting in the validation group. The prediction of bias function matches the real bias function very well which leads to a much more accurate updated model than the original computer model. Table 4.2 gives the quantitative measures of accuracy (validation metrics) of the updated model and the original computer model. These results will be further compared to the one from using Gaussian process model in the next section.

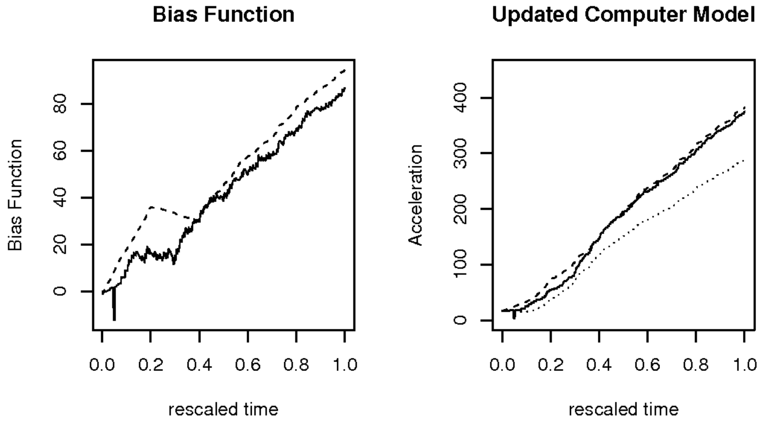


Figure 4.8: Bias Function and Updated Computer Outputs (Prediction: Dashed; Physical (or real bias function): Solid; Computer: Dotted)

4.3.6 Comparison with Gaussian Process Model

As we have mentioned before, several Gaussian Process (GP) approaches have been proposed to model the functional responses of physical and computer experiments. In terms of formulation, a general functional GP approach assumes that

Table 4.2: Validation Metrics for Three Models

	Sprague and Geers Metric
Original Computer Model	0.4836016
Updated GP Model	0.2530137
Updated Functional Regression Model	0.0899152

$$\delta(t) = \sum_{i=1}^{k+1} \beta_i(X) h_i^\delta(t) + \epsilon \tag{4.7}$$

where $\beta_i(X)$'s are GP models and $h_i^\delta(t)$'s are basis functions of time. Higdon et al. (2007) consider $h_i^\delta(t)$ to be the principle component functions and Bayarri et. al. (2007) choose them to be wavelet basis functions. In practice the choice of basis functions should be specific to data type (smooth or irregular, high or low time resolution).

It is our interest to compare the results from the proposed regression-based approach with the existing Gaussian process approaches. Here, we consider the simple extension of the Gaussian process model which uses time as an additional input variable. The sample time points for the GP model are chosen to be 0, 0.2 and 1, which are scaled times. It is a reasonable treatment because the responses within those three points are highly linear with respect to time (see Figure 4.9). Figure 4.10 shows the graphical comparison of those two results. The prediction curve of the functional regression model match the physical outputs better than the curve of the Gaussian process model. Table 4.2 gives the average validation metrics comparing model outputs to physical measurements for different settings in the validation sample. A smaller value of the validation metric indicates better accuracy.

Clearly, the proposed functional regression model achieves better performance in terms of prediction. One reason is that the proposed method uses much more data than the Gaussian Process model, which significantly reduces the prediction variability. In addition, by assuming functional relationship between acceleration and time, the proposed method is better in capturing the intrinsic structure of the

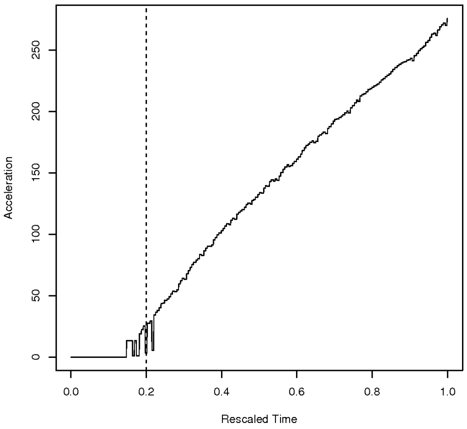


Figure 4.9: A typical functional curve which changes nonlinearly around $t = 0.2$)

observed data.

4.4 Gaussian Process Based Approach

In the previous sections, we have built a functional regression model to analyze functional outputs of physical and computer experiments. It is shown that it can achieve better performance than the Gaussian process model with arbitrarily chosen data due to its inefficiency. However, it is still interesting to develop an approach to improve the Gaussian process model with limited data. Hereafter, we will describe out attempt to this goal and present some results by using our approach.

If the dimensionality of the functional output is small, one might consider either building a separate, independent Gaussian process model for each output quantity (four or five points) or incorporating the variables that index the output spectrum (e.g. time, location) into the Gaussian process model. However, as we described in previous sections, this approach becomes far too cumbersome when there are a large number of time and/or space instances.

One way to tackle this problem of high dimensionality is to use some dimension

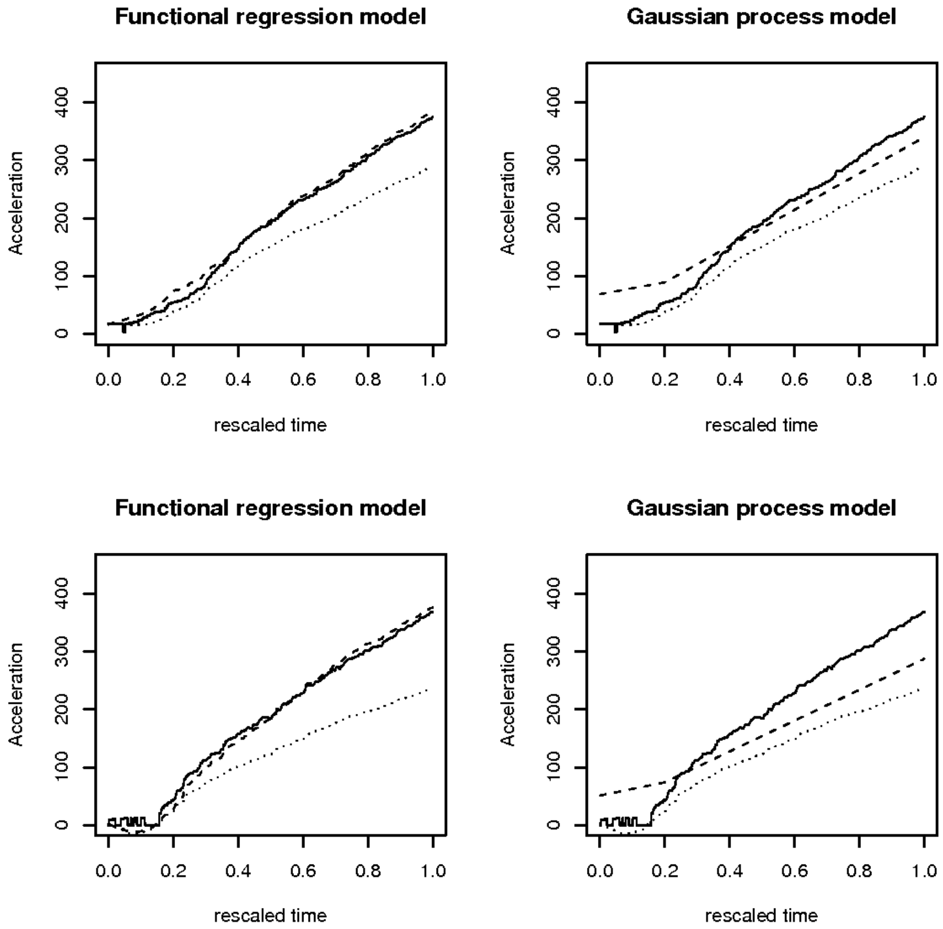


Figure 4.10: Comparison of Functional Regression model (Left) and Gaussian Process model (Right): Top Panel: A: 0 B: 1 C: 1 D: 0.67; Bottom Panel: A: 0 B: 0 C: 0.22 D: 0.67 (Prediction of y^r: Dashed; Physical: Solid; Computer: Dotted)

reduction techniques, such as wavelet decomposition (Bayarri et al., 2007) and principal component analysis (Hidgon et al., 2008). However, these direct extensions to computer experiments have some drawbacks as suggested by Hung et al. (2009). Firstly, it is difficult to interpret results of those functional representations. Secondly, special features of computer experiments are not considered in those techniques (say, deterministic outputs). Lastly, the implementation is not easy for engineers and the learning curve is very deep.

Hung et al. (2009) proposes an approach to naturally extend the kriging model to functional outputs by reducing sampling rates. She utilize the fact that the $N \times N$ correlation matrix Φ can be represented by $\Phi = \Phi(x) \otimes \Phi(t)$ where $\otimes$ is the Kronecker product operation. Since $\Phi^{-1} = \Phi(x)^{-1} \otimes \Phi(t)^{-1}$, the inversion of a $N \times N$ matrix can be replaced by inversions of $n \times n$ and $m \times m$ matrices. Then the problem can be reduced to how to select points for calculating $\Phi(t)^{-1}$. She proposes a point selection scheme to improve uniform selection of time points. The new points are selected to minimize the root mean square prediction error (RMSPE) based on m untried data.

In addition to the point selection scheme, she develops an update formula for $\Phi^{-1}(t)$ to avoid the computational burden of estimating $\Phi^{-1}(t)$ when a new point is added. Assume that the correlation matrix with $k+1$ points is

$$\Phi_{k+1}(t) = \begin{pmatrix} \Phi_k(t) & B \\ B^T & 1 \end{pmatrix},$$

where $B = (\phi(t_1, t_{k+1}), \ldots, \phi(t_k, t_{k+1}))^T$ is a $k \times 1$ vector and $\phi(\cdot, \cdot)$ denotes the correlation function of t. The update formula is defined as

$$\Phi_{k+1}(t)^{-1} = \begin{pmatrix} \Phi_k(t)^{-1} + \Phi_k(t)^{-1} B B^T \Phi_k(t)^{-1} H^{-1} & -\Phi_k(t)^{-1} B H^{-1} \\ -B^T \Phi_k(t)^{-1} H^{-1} & H^{-1} \end{pmatrix} \tag{4.8}$$

where $H = 1 - B^T \Phi_k(t)^{-1} B$. By using this update procedure, the calculation of $\Phi_{k+1}(t)^{-1}$ requires no matrix inversion and thus the computational time can be saved dramatically.

However, this update procedure can not be directly implemented in the GM application since acceleration profiles have irregular time points (not a grid design) after rescaling which invalidates the Kronecker product representation of Φ. In addition, under Ying's procedure, when a new time point is added, all data points sampled at that time are included in the model fitting. Though it is a reasonable procedure, the size of one step becomes too large when the design space is of considerable size (e.g., 128 in the GM example).

McFarland et al. (2008) considers another simple point selection scheme in which one data point is added at a time. The new point is selected as the point with largest prediction error under the existing Gaussian process model. This selection criterion avoids the problem of estimating a new Gaussian process model in each step, however, the selected point is not guaranteed to provide the smallest prediction error when included in the model fitting.

In this section, we will introduce a new procedure which combines ideas from the above two approaches and use the GM example as an illustration.

The new point selection algorithm is as follows.

1. Choose some initial points (Either randomly or using some domain knowledge).

2. Calculate functional prediction error for each run and rank all runs. (Sprague and Geers measure)

3. Starting from the top run, choose one point per run which minimize the functional prediction error until the improvement become negligible or achieve maximum points preset in this step (or overall). This step is the same as Ying's step.

4. Repeat to step 2.

In Step 3, to improve the efficiency of the proposed algorithm, we can also adopt the updating formula developed by Hung et al. (2009). Assume that the correlation

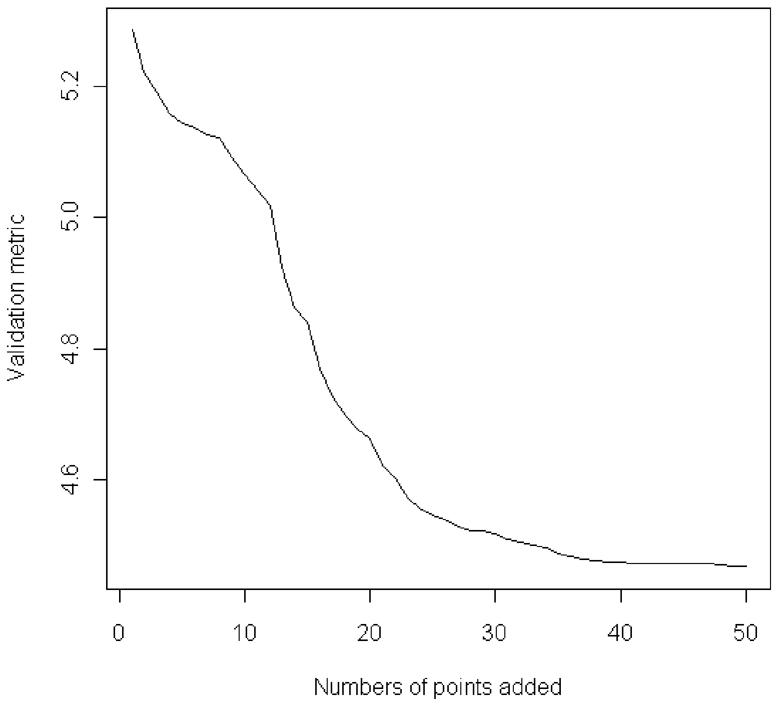


Figure 4.11: Prediction errors versus number of points in the model

matrix with $k+1$ points is

$$\Phi_{k+1}(x,t) = \begin{pmatrix} \Phi_k(x,t) & D \\ D^T & 1 \end{pmatrix},$$

where $D = (\phi_x(x_1, x_{k+1}) \times \phi_t(t_1, t_{k+1}), \ldots, \phi_x(x_k, x_{k+1}) \times \phi_t(t_k, t_{k+1}))^T$ is a $k \times 1$ vector and $\phi_x(\cdot,\cdot)$ and $\phi_t(\cdot,\cdot)$ denotes the correlation function of x and t respectively. Then the update formula 4.8 still holds where let $B = D$. We apply the proposed method to the GM application. The Gaussian process model is defined as

$$Y(x,t) = (x^T, t)\beta + Z(x,t) \tag{4.9}$$

and we use the Gaussian product correlation function

$$c(\mathbf{x}) = \exp(-\sum_{i=1}^{p} \phi_i x_i^2).$$

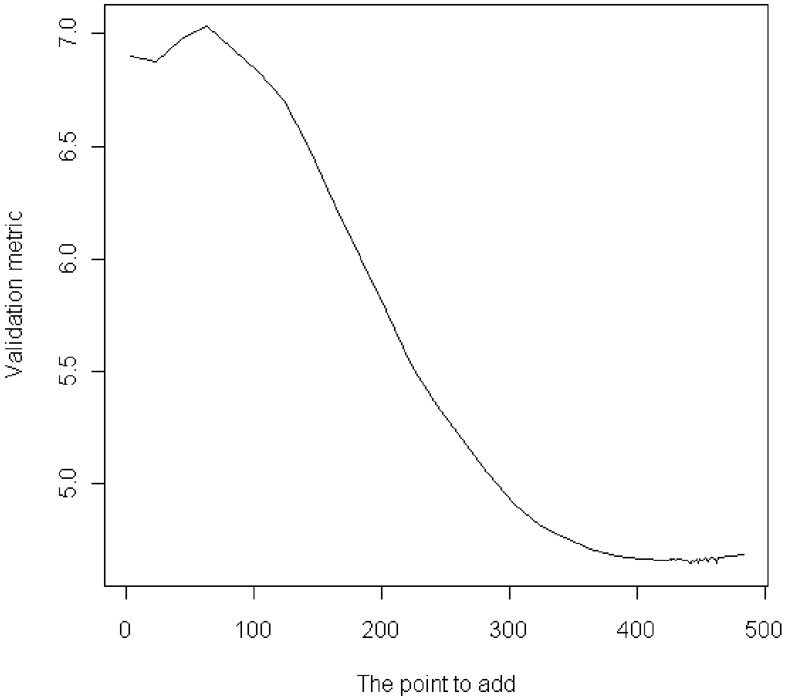


Figure 4.12: Prediction errors versus added time point

The Sprague and Geers metric is considered as our prediction error, since it provides a more comprehensive measure of the difference of two functional responses. The initial points are chosen as starting time points and ending time points of all experimental runs (56 runs). This choice is consistent with the previous analysis.

Figure 4.11 shows the prediction errors as new points are added to our sample. As a comparison, we also calculate the prediction errors (Figure 4.12, the smallest error is 4.645) based on grid design scheme in which we choose 56 points sampled at the same time in one step. We can see that by using our procedures only 20 points are needed to achieve the similar prediction error.

4.5 Concluding Remarks

A functional regression modeling approach is proposed to analyze functional outputs of physical and computer experiments. Traditional procedures for modeling functional data generally involve two steps. Models are first fit for each individual setting to reduce the dimensionality of the functional data, and then the estimated features are treated as new responses, which are further modeled using existing approaches. Alternatively, pointwise models are constructed and then functional curves are fit for parameters estimated pointwisely. On the contrary, the proposed method is more direct and only involves a single step. By reducing the number of steps and following the traditional regression analysis, the proposed model is easier to interpret and implement for practice users. Through a comparison with the existing Gaussian process model, we demonstrate that the proposed method yields sufficient accuracy, performs efficiently and achieves accuracy in global prediction.

In Section 4.5, we introduce a new algorithm to fit the Gaussian process model for functional response. The new algorithm is more useful when a grid design is not presented and we have irregular time points among all experimental runs. In addition, since we can choose different numbers of points for all experimental runs, the new algorithm may have better performance in case that experimental runs show different extents of variations.

CHAPTER V

GAUSSIAN PROCESS MODELING FOR FUNCTIONAL RESPONSES

5.1 Introduction

In Chapter IV, we introduce the functional data problem for computer experiment and modeling. Due to the computational complexity caused by high-dimensionality of the response, it is infeasible to directly apply the traditional Gaussian process modeling approach which requires inversion and determinate calculation. To reduce the computational burden, many different approaches have been proposed for modeling functional data from computer experiments in two steps., such as wavelet decompositions (Bayarri et al., 2007) and principal component analysis (PCA) (Higdon et al., 2007). In our work, we proposed a piecewise linear regression approach for computer model with functional outputs. Since the proposed model is straightforward and only invloves a single step, it is easier to interpret and implement for practice users. In general, the proposed model assumes that

$$Y(t) = f^T(x)\beta(t) + \epsilon(t) \tag{5.1}$$

where $f(x) = [f_1(x), \ldots, f_q(x)]^T$ and $\epsilon(t) \sim N(0, \sigma^2)$. However, we can see that this model is still based on the regular regression model which assumptions can't hold for computer experiment outputs. Thus, in this chapter we will modify our approah to use the Gaussian process assumption rather than the standard normal assumption, which means that

$$Y(t) = f^T(x)\beta(t) + Z(x, t) \tag{5.2}$$

where $Z(x,t)$ is a Gaussian process with mean 0 and the covariance function

$$cov\{Z(x_1,t_1),Z(x_2,t_2)\} = \sigma^2 r(x_1 - x_2, t_1 - t_2).$$

To extend the functional regression model into a functional linear model with error following a Gaussian process as defined above, two main challenges have to be addressed:

1. Variable selection through regularization

   Since the entire variable space $\{f(x,t) = g(x) \times h(t)\}$ is spanned by the design variables and basis functions in the functional dimension, the number of regressors will be considerablely large even if we have a moderate number of design variables and basis functions. Joseph et al. (2008) have argued that unnecessary variables in the mean model can deteriorate the performance. They suggested that only those variables that have a significant effect on the response should be used for the mean model. In this work, we will propose an approach to accomplish variable selection and fitting the kriging model simultaneously.

2. Computational efficiency and irregular grid

   For large N, the calculation of $N \times N$ correlation matrix LU decomposition, inverse and determinant is computationally intensive. Furthermore, the numerical solver for all these operations becomes very unstable when N becomes extremely large due to its rounding errors. Hung et al. (2011) presented a new procedure to overcome the computational problems. It involves using Kronecker product and finding a explicit solution for a part of the correlation matrix. In this paper, we will use the same idea and modified it according to some specific needs in our problem. As we have discussed in Chapter IV, the GM acceleration data have observations at different number of time points for different design combinations (irregular grid). Hung et al. (2011) also presented an idea to treat the observed data as a collection on regular grids but with some 'missing' observations. Then by following an EM-type approach, they will fit the model and estimate the missing data iteratively. The advantages of their approach are that it has very good theoretical properties and is computationally efficient. However, in the GM example, the end time is actually the shift time which implies a systematic change will happen after the end time. Therefore it may not be appropriate to estimate those 'missing' observations after the shift time and use them to fit the model again. In this work, we will provide an simple alternative solution for the irregular grid problem.

This chapter is organized as follows. Section 5.2 presents the formulation of the functional regression kriging model and a literature review of existing approaches. Section 5.3 introduces our proposed model and an algorithm to fit the model. The model's theoretical properties will be given in Section 5.4. In addition to a simulation study, the GM example will be revisted in Section 5.5. We conclude our work in Section 5.6, followed by some future works in Section 5.7.

5.2 Review of Existing Methods

Here we first briefly review the basic formulations for a kriging model and then extent it into a functional kriging model. Several existing approaches will be described and discussed.

Let $Y(x)$ denote a Gaussian process which can be written as

$$Y(x) = f^T(x)\beta + Z(x) \tag{5.3}$$

where $f(x) = (f_1(x), \dots, f_q(x))^T$ is a set of pre-specified functions and $= (\beta_1, ..., \beta_q)^T$ is a set of unknown coefficients. The $Z(x)$ is assumed to be a realization of a stationary Gaussian process with covariance

$$cov(Z(x), Z(x^*)) = \sigma^2 R(x, x^*) = \sigma^2 \exp[-d(x, x^*)]. \tag{5.4}$$

The correlation function $R(x, x^*)$ in (2) is a function of the "distance" between x and x^* . Many correlation functions are available in the literature (Santner et al. (2003)). In this research, we will consider the Gaussian correlation function which implies the distance $d(x, x^*) = \sum \phi_i (x^i - (x^*)^i)^2$ for design variables and $d(t, t^*) = \phi_i |t - t^*|$ for the functional space.

As we have described in Chapter IV, usually functional responses are collected over an interval of an index denoted by t. For example, under the setting x_i, the functional response $y_i(t)$ is observed at discrete time points $t_{i1}, \ldots, t_{im_i}$. A direct extension of kriging model to functional responses will incorporate the functional index (t) as an additional input of the model. Thus, the functional kriging regression model will be

$$Y(x, t) = f^T(x, t)\beta + Z(x, t) \tag{5.5}$$

where $Y(x, t)$ is the response measured at point t in the functional space given the input variable x. $f(x, t) = [f_0(x, t), \ldots, f_q(x, t)]^T$ are collections of known functions and usually $f_0(x, t) = 1$. We assume that $Z(x, t)$ is a Gaussian process with mean 0 and the covariance function $cov(Z(x, t), Z(x^*, t^*)) = \sigma^2 R((x, t), (x^*, t^*))$. Usually, a separable product correlation struction is assumed for the correlation function. That is, $R((x, t), (x^*, t^*)) = R_x(x, x^*) R_t(t, t^*)$, where $R_x(x, x^*)$ is the same as we have defined in the standard kriging model and $R_t(t, t^*)$ is the correlation function for variable t. Correlation parameters associated with these correlation functions are denoted by $\boldsymbol{\phi}$.

Suppose the $N \times 1$ vector $y = (\mathbf{y_1}^\mathbf{T}, \ldots, \mathbf{y_n}^\mathbf{T})^\mathbf{T} = (\mathbf{Y_1}, \ldots, \mathbf{Y_N})^\mathbf{T}$ is the collection of all the outputs with $N = \sum_1^n m_i$ and (X_i, T_i) is associated with the output y_i. Based on model (5.5), the universal kriging predictor is given by

$$\hat{y}(x, t) = f(x, t)^T \hat{\mu} + R_0(x, t)^T R_N^{-1}(y - F\hat{\mu}) \tag{5.6}$$

where $F = (f(X_1, T_1), \ldots, f(X_N, T_N))^T$, $\hat{\mu} = F^T R_N^{-1} y / F^T R_N^{-1} F$,

$R_0(x,t) = (R((x,t),(X_1,T_1)),\ldots,R((x,t),(X_N,T_N)))^T$ and R_N is an $N \times N$ matrix with elements $R_N(i,j) = R((X_i,T_i),(X_j,T_j))$. The correlation parameters are estimated by minimizing the negative log-likelihood

$$\hat{\boldsymbol{\phi}} = \arg\max_{\boldsymbol{\phi}}(N\log\hat{\sigma}^2 + \log|R_N|), \tag{5.7}$$

where $\hat{\sigma}^2 = \frac{1}{N}(y - F\hat{\mu})^T R_N^{-1}(y - F\hat{\mu})$. However, Hung et al. (2011) states that the direct implementation of this model is computationally prohibitive due to the inversion and determinant calculations of the covariance matrix (R_N^{-1} and $|R_N|$). To tackle the computational difficulty associated with functional response and successfully extend kriging, they proposed a new procedure in their work. For our problem, we intend to follow their approach in reducing the computational difficulty and a few modifications have to be made to accomodate some special needs of this particular data.

Another type of approach which we will compare to in this chapter is to first apply dimension reduction methods, e.g. PCA or SVD, to functional data. Then independent Gaussian processes are fitted based on the results of dimension reduction methods. The model will combine these two steps together to estimate and predict. Techniques such as wavelet decomposition (Bayarri et al. 2007) and principal component analysis (PCA) (Higdon et al. 2007) have been used to model functional outputs. Here we describe the model setup from Dancik et al. (2008). Let $[y]_{ij}$ be a $m \times n$ matrix with elements $Y_j(t_i)$ and $r = \min(m,n)$. Using singular value decomposition,

$$[y]_{ij} = U_{m\times r}D_{r\times r}V^T_{r\times n} = \sum_{p=1}^{r}\lambda_p\{\alpha_p\}_i\{w_p(x)\},$$

where λ_p is the p^{th} singular value, α_p is the p^{th} column of U, and $w_p(x)$ is the p^{th} row of V^T. The j^{th} column of V^T, which contains the elements $\{w_p(x)\}$, $p = 1,\ldots,r$, is called a vector of principle component weights corresponding to the j^{th} design setting. The output Y is approximated by keeping the $l < r$ most important principle component weights, corresponding to the l largest singular values. Then independent

Gaussian processes are fitted to the most important principle component weights $w_1(x), \ldots, w_l(x)$.

5.3 Models and Methods of Analysis

5.3.1 The functional regression model with Gaussian process errors

In functional data analysis (FDA), the functional regression can correspond to either functional predictors or functional responses. James et al. (2009) proposed a functional linear regression model for functional predictors which, through variable selection techniques, produce estimates that are both interpretable, flexible and accurate. In our work, we will extend their model to both scalar and functional responses with Gaussian process error.

The standard functional regression (FLR) model relates functional predictors to a scalar response via

$$Y_i = \beta_0 + \int X_i(t)\beta(t)dt + \epsilon_i, \quad i = 1, \ldots, n \tag{5.8}$$

where $\beta(t)$ is the "coefficient function". There are generally two approaches for modeling the structure of $\beta(t)$ to interpolate the responses. In the first method, $\beta(t)$ is represented using a p-dimensional basis function, $\beta(t) = B(t)^T\eta$ where p is expected to be large enough to capture the patterns in $\beta(t)$ but small enough to regularize the fit. Then (5.8) can be rewritten as $Y_i = \beta_0 + \boldsymbol{X}_i^T\eta + \epsilon_i$, where $\boldsymbol{X}_i = \int X_i(t)\boldsymbol{B}(t)dt$, and η can be estimated using ordinary least squares. This method actually is directly related to the approach we have developed in Chapter IV, with the exception that we are dealing with functional responses rather than scalar responses. In the second method, a penalized least squares estimation procedure is introduced to shrink variability in $\beta(t)$. The penalty, $P(\beta)$ can be either of the form $\int \beta^{(d)}(t)^2dt$ with $d = 2$ being a common choice or of the form $\|\eta\|_1$. In either cases, $\beta(t)$ can be found by minimizing $\sum_{i=1}^n (Y_i - \beta_0 - \int X_i(t)\beta(t)dt)^2 + \lambda P(\beta)$.

Based on these existing results, James et al. (2009) presented a new idea to reformulate the problem as a form of variable selection. In particular they divide the time period up into a fine grid of points. Then they use variable selection methods to determine whether the dth derivative of $\beta(t)$ is zero or not at each of the grid points. The formulation of their method is described as follows.

First they assume there is a p-dimensional basis function $\boldsymbol{B}(t) = [b_1(t), \ldots, b_p(t)]^T$ and

$$\beta(t) = \boldsymbol{B}(t)^T \eta + e(t) \tag{5.9}$$

, where $e(t)$ represents the deviations of the true $\beta(t)$ from their model. They also showed that since p can be chosen arbitrarily large under this method, $|e(t)|$ can generally be assumed to be small.

Combining (5.8) and (5.9) we will have

$$Y_i = \beta_0 + \boldsymbol{X}_i^T \eta + \epsilon_i^* \tag{5.10}$$

where $\boldsymbol{X}_i = \int X_i(t) \boldsymbol{B}(t) dt$ and $\epsilon_i^* = \epsilon_i + \int X_i(t) e(t) dt$.

Let $t_1, \ldots, t_p$ stands for p evenly spaced points in the functional space, approximation of $\beta^{(2)}(t_j)$ can be written as $\gamma_j = p^2 \left[B(t_j)^T \eta - 2B(t_{j-1})^T \eta + B(t_{j-2})^T \eta \right]$. Therefore, we have

$$\gamma = A\eta,$$

where $A = \left[D^d B(t_1), \ldots, D^d B(t_p) \right]^T$, and D^d is the dth finite difference operator where $DB(t_j) = p \left[B(t_j)^T - B(t_{j-1})^T \right]$, $D^2 B(t_j) = p^2 \left[B(t_j)^T \eta - 2B(t_{j-1})^T \eta + B(t_{j-2})^T \eta \right]$, etc. Let $V = [1|XA^{-1}]$, the final model can be written as

$$Y = V\gamma + \epsilon^*. \tag{5.11}$$

If we replace the standard normal error ϵ^* with a Gaussian process error Z, the kriging version of the functional regression model for scalar responses will be

$$Y = V\gamma + Z \tag{5.12}$$

where $Z \sim N(0, \Sigma)$ and $\Sigma(i,j) = R(X_i(t), X_j(t))$. A common choice for R will be $R(x(t), y(t)) = \exp(-\int \phi |x(t) - y(t)|^\rho dt)$.

To make this approach work for functional responses, we first have to rewrite the model. The basic model states that

$$Y_i(t) = f^T(\boldsymbol{X}_i)\boldsymbol{\beta}(t) + \epsilon_i(t) \tag{5.13}$$

where $\boldsymbol{X}_i = (x_{i1}, \ldots, x_{is})$, $f(\boldsymbol{X}_i) = F_i = (f_1(\boldsymbol{X}_i), \ldots, f_q(\boldsymbol{X}_i))^T$ and $\boldsymbol{\beta}(t) = (\beta_1(t), \ldots, \beta_q(t))^T$. It should be noted that in general, $\boldsymbol{X}_i$ could also change over time which means $\boldsymbol{X}_i(t)$. However, in this thesis, we will only consider the case that $\boldsymbol{X}_i$ is constant over time. And it is not difficult to extend our results to time-varing $\boldsymbol{X}_i$. Based on the same basis function we have defined above, we have that $\beta_j(t) = B(t)^T\boldsymbol{\eta}_j + e_j(t)$ for $j = 1, \ldots, q$. Let $\boldsymbol{\eta} = (\eta_{11}, \ldots, \eta_{1p}, \eta_{21}, \ldots, \eta_{qp})^T$ and $\boldsymbol{e}(t) = (e_1(t), \ldots, e_q(t))^T$, the regression model can be rewritten as

$$Y_i(t) = (f(\boldsymbol{X}_i) \otimes B(t))^T\boldsymbol{\eta} + \epsilon_i^*(t) \tag{5.14}$$

where $\epsilon_i^*(t) = f^T(\boldsymbol{X}_i)\boldsymbol{e}(t) + \epsilon_i(t)$. Then let

$$\boldsymbol{Y} = (Y_1(t_{11}), \ldots, Y_1(t_{1m}), Y_2(t_{21}), \ldots, Y_n(t_{nm}))^T,$$

$$\boldsymbol{B} = (B(t_1), \ldots, B(t_m))$$

and

$$\boldsymbol{F} = (F_1, F_2, \ldots, F_n),$$

the final model will be

$$\boldsymbol{Y} = \boldsymbol{W}^T\boldsymbol{\eta} + \epsilon^* \tag{5.15}$$

where $\boldsymbol{W} = \boldsymbol{F} \otimes \boldsymbol{B}$.

By choosing some particular basis functions, we could assume that $\boldsymbol{\eta}$ is sparse and estimate $\boldsymbol{\eta}$ using a variable selection procedure such as Lasso. If there is no reason to assume that $\boldsymbol{\eta}$ will be sparse, we can also adopt the idea from James et al. (2009)

which assumes that one or more of its derivatives are sparse i.e. $\beta^{(d)}(t) = 0$ over large regions of t for one or more values of $d = 0, 1, 2, \ldots$. Then let $\boldsymbol{A} = I_{q\times q} \otimes A$ and $\boldsymbol{\gamma} = (\gamma_{11}, \ldots, \gamma_{1p}, \gamma_{21}, \ldots, \gamma_{qp})^T$, if

$$\boldsymbol{\gamma} = \boldsymbol{A}\boldsymbol{\eta},$$

the alternative model will be

$$\boldsymbol{Y} = \boldsymbol{V}^T\boldsymbol{\gamma} + \epsilon^* \tag{5.16}$$

where $\boldsymbol{V} = \boldsymbol{W}\boldsymbol{A}^{-1}$. Like the model for scalar responses, p can be chosen arbitrarily large here. Thus $f^T(\boldsymbol{X}_i)\boldsymbol{e}(t)$ can also be assumed to be small. We should note that if we believe more than one derivatives of $\boldsymbol{\beta}(t)$ are sparse, $\boldsymbol{A}$ may no longer be invertible. In that case, James et al. (2009) provided one useful extension which can be directly applied in such a problem.

Again here we could replace the error ϵ^* with a Gaussian process error Z. Then the kriging version of the functional regression model for functional responses will be

$$\boldsymbol{Y} = \boldsymbol{W}^T\boldsymbol{\eta} + Z \text{ or } \boldsymbol{Y} = \boldsymbol{V}^T\boldsymbol{\gamma} + Z \tag{5.17}$$

where $Z \sim N(0, R_N)$.

The kriging predictor is given by

$$\hat{y}(\boldsymbol{x}, t) = (f(\boldsymbol{x}) \otimes \boldsymbol{B}(t))^T\hat{\boldsymbol{\eta}} + R_0(\boldsymbol{x}, t)^T R_N^{-1}(\boldsymbol{Y} - \boldsymbol{W}^T\hat{\boldsymbol{\eta}}) \tag{5.18}$$

where $\hat{\boldsymbol{\eta}}$ are the Lasso estimate under some penalty paramter λ. In the next section, we will explain how to find the Lasso estimate.

5.3.2 Fitting the Model

Since η or γ is assumed sparse, if the model has the standard Normal errors, potentially a variety of variable selection methods can be used to fit (5.11) and (5.15). A few examples include the Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), the Elastic Net (Zou and Hastie, 2005) and Danzig selector (Candes and Tao, 2007). We opted to focus on the Lasso selector in this work since it has demonstrated strong empirical results on variable selection and also has very efficient algorithm to calculate the solution path. However, it will not be difficult to extend our approach to other similar variable selection methods.

In the linear regression model $Y = X\beta + \epsilon$, the Lasso estimate, $\hat{\beta}$, is defined by

$$\hat{\beta}_L = \arg\min_{\beta} \|Y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{5.19}$$

where $\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1$ is the penalized negative log-likelihood, $\|\cdot\|_1$ and $\|\cdot\|_2$ respectively denote the L_1 and L_2 norms and $\lambda \geq 0$ is a tuning parameter.

Under the kriging model, the penalized negative log-likelihood becomes

$$\begin{aligned} L(\sigma, \boldsymbol{\eta}, \theta|Y) &= N\log(\sigma^2) + \log(|R_N(\theta)|) \\ &+ \frac{1}{\sigma^2}(\boldsymbol{Y} - \boldsymbol{W}^T\boldsymbol{\eta})^T R_N(\theta)^{-1}(\boldsymbol{Y} - \boldsymbol{W}^T\boldsymbol{\eta}) + \lambda\|\boldsymbol{\eta}\|_1 \end{aligned} \tag{5.20}$$

The regression coefficents $\boldsymbol{\eta}$, the variance parameter σ and the correlation parameters θ are estimated by minimizing $L(\sigma, \boldsymbol{\eta}, \theta|Y)$, i.e.

$$(\hat{\sigma}, \hat{\boldsymbol{\eta}}, \hat{\theta}) = \arg\min_{\sigma,\boldsymbol{\eta},\theta} l(\sigma, \boldsymbol{\eta}, \theta|Y).$$

Given $\boldsymbol{\eta}$ and θ, we have

$$\hat{\sigma}^2 = \frac{1}{N}(\boldsymbol{Y} - \boldsymbol{W}^T\boldsymbol{\eta})^T R_N(\theta)^{-1}(\boldsymbol{Y} - \boldsymbol{W}^T\boldsymbol{\eta}).$$

Li and Sudjianto (2005) considered an iterative procedure to estimate η and θ. Though the penalized parameter is θ, in their model, we could still adopt a similar scheme for estimation of our model. The two iterative steps we consider here are based on two conditions: when θ and σ are given and when η is given.

In the first case (step A), after comparing (5.19) and (5.20), we can show that for given values of θ, the Lasso model and the penalized kriging model are equivalent. Thus, $\hat{\eta}$ can be estimated through the efficient algorithm for Lasso. To reexpress

(5.20), we have to use the fact that $R_N(\theta) = U^T U$ by calculating the LU decomposition of $R_N(\theta)$ where U is an upper triangle matrix. Then (5.20) becomes

$$\begin{aligned} L(\sigma, \eta, \theta|Y) &= N \log(\sigma^2) + \log(|U^T U|) \\ &+ \frac{1}{\sigma^2}(\boldsymbol{Y} - \boldsymbol{W}^T \boldsymbol{\eta})^T R_N(\theta)^{-1}(\boldsymbol{Y} - \boldsymbol{W}^T \boldsymbol{\eta}) + \lambda \|\eta\|_1 \end{aligned}$$

Since θ and σ are given, to optimize $l(\boldsymbol{\eta}, \theta, \sigma|\boldsymbol{Y})$ over $\boldsymbol{\eta}$ we can consider the part involving only $\boldsymbol{\eta}$ which is:

$$L_1(\boldsymbol{\eta}|\boldsymbol{Y}, \sigma, \theta) = \frac{1}{\sigma^2}(\boldsymbol{Y} - \boldsymbol{W}^T \boldsymbol{\eta})^T (U^T U)^{-1}(\boldsymbol{Y} - \boldsymbol{W}^T \boldsymbol{\eta}) + \lambda \|\boldsymbol{\eta}\|_1.$$

Let $\boldsymbol{Y}_t = (U^{-1})^T \boldsymbol{Y}/\sigma$ and $\boldsymbol{W}_t = \boldsymbol{W} U^{-1}/\sigma$, then

$$L_1(\boldsymbol{\eta}|\boldsymbol{Y}, \sigma, \theta) = \|\boldsymbol{Y}_t - \boldsymbol{W}_t^T \eta\|_2^2 + \lambda \|\boldsymbol{\eta}\|_1 \tag{5.21}$$

which means that $\boldsymbol{Y}_t$ and $\boldsymbol{W}_t$ can be used by existing Lasso algorithms to estimate $\hat{\eta}$.

In the latter case (step B), taking out the penalty term from the penalized kriging model, it becomes the standard kriging model. For σ, we have

$$\hat{\sigma}^2 = \frac{1}{N} \|\boldsymbol{Y}_t - \boldsymbol{W}_t^T \boldsymbol{\eta}\|_2^2.$$

For θ, it is estimated by minimizing $N \log(\hat{\sigma}^2) + \log(|R_N(\theta)|)$. Any optimization algorithm would be a feasible solution for this problem. To speed up the method, we could also adopt the Fisher scoring algorithm by Li and Sudjianto (2005).

The computing algorithm for the proposed procedure is summarized as follows.

1. Choose initial values for $\hat{\boldsymbol{\eta}}^{(0)}$,$\hat{\theta}^{(0)}$,$\hat{\sigma}^{(0)}$. $\hat{\boldsymbol{\eta}}^{(0)}$ can be chosen by fitting the functional regression model (5.15) directly.

2. Perform step A to estimate $\hat{\boldsymbol{\eta}}^{(i+1)}$ using the LARS algorithm. λ is chosen by minimizing the C_p value.

3. Perform step B to estimate $\hat{\theta}^{(i+1)}$,$\hat{\sigma}^{(i+1)}$ by using $\hat{\boldsymbol{\eta}}^{(i+1)}$.

4. Repeat 2 and 3 until the convergence is achieved. Then declare $\hat{\boldsymbol{\eta}}$,$\hat{\theta}$ and $\hat{\sigma}$ to be the estimates.

In the second step, we use the C_p criterion instead of the more popular cross validation because the transformed matrix $\boldsymbol{W}_t$ can't be directly split into training and testing groups. An algorithm using cross validation would be formulated as follows.

1. Choose initial values for $\hat{\boldsymbol{\eta}}^{(0)}$,$\hat{\theta}^{(0)}$,$\hat{\sigma}^{(0)}$. $\hat{\boldsymbol{\eta}}^{(0)}$ can be chosen by fitting the functional regression model (5.15) directly.
2. Randomly split the training set in to k groups.
3. Calculate $\boldsymbol{W}_t^{(-j)}$ for the training data except the jth group for $j = 1, \ldots, k$.
4. Perform step A to estimate $\hat{\boldsymbol{\eta}}^{(-j)(i+1)}$ using the LARS algorithm for all λ and $j = 1, \ldots, k$.
5. Perform step B to estimate $\hat{\theta}^{(-j)(i+1)}$,$\hat{\sigma}^{(-j)(i+1)}$ by using $\hat{\boldsymbol{\eta}}^{(-j)(i+1)}$ for all λ and $j = 1, \ldots, k$.
6. Repeat 3, 4 and 5 until the convergence is achieved. Then λ is chosen to minimize the validation error and the corresponding $\hat{\boldsymbol{\eta}}$,$\hat{\theta}$ and $\hat{\sigma}$ are declared to be the estimates.

5.3.3 Implementation Details

Here we will discuss more details of implementing the previous algorithm for the GM acceleration data. Especially, we will follow the work of Hung et al. (2011) to develop efficient algorithm for matrix operations.

5.3.3.1 Efficient Matrix Operations

Though the algorithm developed in the last section enables us to estimate all parameters η, θ and σ, the calculation of $N \times N$ correlation matrix LU decomposition, inverse and determinant is still computationally intensive. Furthermore, the numerical solver for all these operations becomes very unstable when N becomes extremely large due to its rounding errors. Therefore, it is necessary to develop an explicit solution rather than relying on the numerical solution. The idea in Hung et al. (2011) is to take advantage of the product form of the correlation function together with the Kronecker product techniques. By using these, explicit solutions of those matrix operations can be derived.

Assume that the functional data are collected with regular grid, the functional responses are observed in the same locations for each experimental setting, i.e., $t_{1j} = \cdots = t_{nj} = t$ for all $j = 1, \ldots, m$. Then the correlation matrix $R_N(\theta)$ can be represented by $R_N = R_x \otimes R_t$, where $\otimes$ refers to the Kronecker product operation, R_x us a n-by-n correlation matrix with elements $R(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and R_t is a m-by-m correlation matrix with elements $R(t_i, t_j)$. Under the Kronecker product form, R_N^{-1} can be calculated as

$$R_N^{-1} = R_x^{-1} \otimes R_t^{-1},$$

$|R_N|$ will be

$$|R_N| = m|R_x| \cdot n|R_t|$$

and the upper matrix of the LU decompostion U_N will be

$$U_N = U_x \otimes U_t$$

where U_x and U_t are upper matrixs of the LU decompostion of R_x and R_t respectively. It is obvious that those matrix operations on R_x and R_t are much cheaper and more stable than the ones of $N \times N$ matrix R_N.

However, for the functional model we are considering here, m could still be quite large for numerical program to handle due to its near singularity. It motivates us to consider a special case of the regular grid collection, under which observations are collected with equal spacing, i.e., $t_{i2} - t_{i1} = \cdots = t_{im} - t_{i(m-1)} = \Delta t$ for all

$i = 1, \ldots, n$. For the purpose of illustration, we assume that the colletion points in the functional space are $1, \ldots, m$ for each experimental setting. In this situation, the computational burden in fitting the kriging model can be further reduced by a special correlation matrix which results closed-form solutions for LU decompostion, inversion and determinant of the correlation matrix.

As defined before, we have $R_t(i, j) = \exp[-d_t(i, j)] = \exp[-\theta_t|i-j|]$. Let $\rho = e^{-\theta_t}$, then $R_t(i, j) = \rho^{|i-j|}$ and R_t can be written as

$$R_t = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{m-1} \\ \rho & 1 & \rho & \cdots & \rho^{m-2} \\ \vdots & \vdots & \vdots & & \vdots \\ \rho^{m-1} & \rho^{m-2} & \cdots & \cdots & 1 \end{bmatrix}.$$

By some calculations, we can show that $|R_t| = (1 - \rho^2)^{m-1}$,

$$U_t = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{m-1} \\ 0 & (1-\rho^2)^{1/2} & \rho(1-\rho^2)^{1/2} & \cdots & \rho^{m-2}(1-\rho^2)^{1/2} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & (1-\rho^2)^{1/2} & \rho(1-\rho^2)^{1/2} \\ 0 & 0 & 0 & 0 & (1-\rho^2)^{1/2} \end{bmatrix} \tag{5.22}$$

and

$$U_t^{-1} = \frac{1}{(1-\rho^2)^{1/2}} \begin{bmatrix} (1-\rho^2)^{1/2} & -\rho & 0 & \cdots & 0 \\ 0 & 1 & -\rho & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 1 & -\rho \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.23}$$

Since U_t can be written in a closed form and U_x is generally easy to calculate, the complexity of fitting the kriging model (5.17) becomes much smaller.

5.3.3.2 Replicates in Computer Experiments

As we described in the last chapter, a special property of the GM experiment is that there are other uncontrollable input variables in the computer model which generate two replicate observations for each design setting. The model we introduced in the precious sections can only deal with the situation in which there is exactly one observation for each combination of a design setting and a point in the functional space. To overcome this problem, first we will reexpress our model (5.13) to handle replicate observations. A functional kriging regression model with replicates is given by

$$Y_{ik}(t) = f^T(\boldsymbol{X}_i)\boldsymbol{\beta}(t) + Z_i(t) + \zeta_{ik}(t) \quad i = 1 \dots n, k = 1, \dots d, \tag{5.24}$$

where $Y_{ik}(t)$ and $\zeta_{ik}(t)$ are the observation and the random effect or error function respectively, for design setting i and replicate k.

In literature, model (5.24) can be generally considered to be a two-level hierachical model where units (replicates) are nested within treatment groups (design settings). There have been some developments of methodologies for dealing wih hierarchical, spatially correlated functional data. Most recently, Zhou et al. (2010) develop a reduced rank mixed effects model under which the unit level random effects are modeled using different sets of principal components and the spatial correlation of unit level random functions is modeled through the spatial correlation of the principal component scores. In our situation, since the main objective is to study the mean function $f^T(\boldsymbol{X})\boldsymbol{\beta}(t)$, we will adopt a simplified assumption on $\zeta_{ik}(t)$. However, it is possible to extend their approach to the functional kriging regression model as the hierarchy of data becomes more complex.

In the functional kriging model with replicates, we assume that $\zeta_{ik}(t)$'s are mean 0 and independent with $Z_i(t)$'s. The covariance $cov(\zeta_{ik}(t), \zeta_{i'k'}(t')) = 0$ if $i \neq i'$ or $k \neq k'$ and $cov(\zeta_{ik}(t), \zeta_{ik}(t')) = \sigma_\zeta^2 \exp(-\theta_\zeta |t - t'|)$. Then the covariance matrix of $\boldsymbol{Y}$

in model (5.17) becomes

$$\begin{aligned} cov(\boldsymbol{Y}, \boldsymbol{Y}) &= cov(\boldsymbol{W}^T\boldsymbol{\eta} + Z + \zeta, \boldsymbol{W}^T\boldsymbol{\eta} + Z + \zeta) \\ &= cov(Z + \zeta, Z + \zeta) \\ &= cov(Z, Z) + cov(\zeta, \zeta) \\ &= \sigma_Z^2 R_N + \sigma_\zeta^2 R^\zeta \end{aligned}$$

where $R_N = R_x \otimes R_t$ and $R^\zeta = I_{n\times n} \otimes R_t^\zeta$.

To further simplify the calculation in our problem, we will assume that $\theta_t = \theta_\zeta$. This is a reasonable assumption as long as the replicate functional responses from the same design setting have similar shapes in the functional space. Under this assumption, we have $R_t^\zeta = R_t$ and

$$cov(\boldsymbol{Y}, \boldsymbol{Y}) = (\sigma_Z^2 R_x + \sigma_\zeta^2 I_{n\times n}) \otimes R_t.$$

Then let

$$\Sigma_x = R_x + (\sigma_\zeta/\sigma_Z)^2 I_{n\times n},$$

$cov(\boldsymbol{Y}, \boldsymbol{Y})$ can be written as

$$cov(\boldsymbol{Y}, \boldsymbol{Y}) = \sigma_Z^2 \Sigma_N = \sigma_Z^2(\Sigma_x \otimes R_t).$$

Assume U is the upper triangle matrix of the LU decompostion of Σ_N, given θ and η σ_Z can be estimated the same way as before,

$$\hat{\sigma_Z}^2 = \frac{1}{N}\|Y_t - W_t^T\eta\|_2^2.$$

5.3.3.3 Irregular Grid and Rescaled Time

As discussed in the last chapter, the GM acceleration data have observations at different numbers of time points for different design combinations. We also rescale $[0, t_{shift}]$ to $[0, 1]$ by dividing each time point, t, by the shift time t_{shift} to effectively predict both functional responses and the shift time. Under the original time scale, we could utilize the EM algorithm proposed by Hung et al. (2011) to overcome the irregular grid problem. However, in this work, we plan to exploit an alternative strategy for the rescaled time.

Since all observations fall into the unit interval $[0, 1]$ under the rescaled time, we can devide $[0, 1]$ into m-1 equal subintervals, i.e., $[0, 1/m)$,. . ., $[\frac{m-1}{m}, 1]$. Using the collected observations, we can interpolate on those boundary points, 0, $1/m$, . . ., 1. Since we have an intensive sampling rate and assume the functional responses are smooth, the loss of information would be almost negligible as long as m is reasonably large. We could also take one further step to update the interpolated values using estimated parameters. The procedures will continue until the convergence or no improvement.

5.4 Theoretical Properties

In this section, we will show that the penalized functional kriging model enjoys similar good theoretical properties as the FLiRTI approach in James et.al (2009) which include tight, non-asymptotic, bounds on the error in the estimate.

Let $\hat{\eta}_\lambda$ correspond to the Lasso solution using tuning parameter λ. Let D_λ be a diagonal matrix with jth diagonal equal to 1, -1 or 0 depending on whether the jth component of $\hat{\eta}_\lambda$ is positive, negative or zero respectively. Consider the following condition on the transformed design matrix, W_t,

$$\boldsymbol{u} = (D_\lambda \tilde{W}_t(\theta)\tilde{W}_t(\theta)^T D_\lambda)^{-1}\mathbf{1} \geq \mathbf{0} \quad \text{and} \quad \left\|\tilde{W}_t(\theta)\tilde{W}_t(\theta)^T D_\lambda \boldsymbol{u}\right\|_\infty \leq 1, \tag{5.25}$$

where $\tilde{W}_t$ corresponds to W_t after standardizing its rows, $\mathbf{1}$ is a vector of ones and the inequality for vectors is understood componentwise. (5.25) is required for the Lasso to enjoy the Dantzig selector's non-asymptotic bounds (Candes and Tao, 2007). The following Theorem 1 is an extension of Theorem 1 in James et.al (2009).

Theorem 1 *For a given p-dimensional basis $\boldsymbol{B}_p(t)$, let $\omega_{ip} = \sup_t |e_{ip}(t)|$ for each $i = 1, \ldots, q$. Suppose that η_p has at most S_{pq} non-zero components and $\delta^W_{2S_{pq}}$ +*

$\phi^W_{S_{pq},2S_{pq}} < 1$. *Further, suppose that we estimate* $\boldsymbol{\beta}(t)$ *using the penalized functional kriging model with any value of* λ *such that (5.25) holds and*

$$\max |\tilde{W}_t(\theta)\epsilon^*| \leq \lambda. \tag{5.26}$$

Then, for every $0 \leq t \leq 1$,

$$|\hat{\beta}_i(t) - \beta_i(t)| \leq \frac{1}{\sqrt{nm}} C^\theta_{n,p,i}(t)\lambda\sqrt{S_{pq}} + \omega_{ip} \tag{5.27}$$

for each $i = 1, \ldots, q$.

As suggested by James et.al (2009), the constant δ and ϕ are both measures of the orthogonality of W. The closer they are to zero the closer W is to orthogonal. The condition $\delta^W_{2S_{pq}} + \phi^W_{S_{pq},2S_{pq}} < 1$ ensures that $\boldsymbol{\beta}(t)$ is identifiable.

Before we prove the theorem, we first present definitions of δ, ϕ and $C^\theta_{n,p,k}(t)$.

Definition 1 *Let* X *be an* $n \times m$ *by* $p \times q$ *matrix and let* X_T, $T \subset \{1, \ldots, pq\}$ *be the* $n \times m$ *by* $|T|$ *submatrix obtained by standardizing the columns of* X *and extracting those corresponding to the indices in* T. *Then we define* δ^X_S *as the smallest quantity such that* $(1 - \delta^X_S) \|\boldsymbol{c}\|_2^2 \leq \|X_T\boldsymbol{c}\|_2^2 \leq (1 + \delta^X_S) \|\boldsymbol{c}\|_2^2$ *for all subsets* T *with* $|T| \leq S$ *and all vectors* $\boldsymbol{c}$ *of length* $|T|$.

Definition 2 *Let T and T' be two disjoint sets with* $T, T' \subset \{1, \ldots, pq\}$, $|T| \leq S$ *and* $|T'| \leq S$. *Then, provided* $S + S' \leq pq$, $\phi^X_{S,S'}$ *is defined as the smallest quantity such that* $(X_T\boldsymbol{c})^T X_{T'}\boldsymbol{c}' \leq \phi^X_{S,S'} \|\boldsymbol{c}\|_2 \|\boldsymbol{c}'\|_2$ *for all T and T' and all corresponding vectors* $\boldsymbol{c}$ *and* $\boldsymbol{c}'$.

Finally, let $C^\theta_{n,p,k}(t) = \dfrac{\alpha^\theta_{n,p,k}(t)}{1 - \delta^W_{2S} - \phi^W_{S,2S}}$ where $\alpha^\theta_{n,p,k}(t) = \sqrt{\sum_{i \in T_k} \dfrac{b_i^2(t)}{\frac{1}{nm}\sum_{j=1}^{nm} w_{ij}(\theta)^2}}$ and T_k is the indices corresponding to $f_k(\cdot)$ for $k = 1, \ldots, q$.

The following lemma comes from James et.al (2009) and is utilized in the proof of Theorem 1.

Lemma 1 *Let* $\boldsymbol{Y} = \tilde{X}^T\tilde{\eta} + \epsilon$ *where* $\tilde{X}$ *has norm one columns. Suppose that* $\tilde{\eta}$ *is an S-sparse vector with* $\delta^X_{2S} + \phi^X_{S,2S} < 1$*. Let* $\hat{\tilde{\eta}}$ *be the corresponding solution from the Lasso. Then* $\left\|\hat{\tilde{\eta}} - \tilde{\eta}\right\| \leq \frac{4\lambda\sqrt{S}}{1-\delta^X_{2S}-\phi^X_{S,2S}}$ *provided that (5.25) and* $\max|\tilde{X}\epsilon| \leq \lambda$ *both hold.*

Now we begin the proof of Theorem 1. First note that functional regression model for kriging errors can be reexpressed as,

$$\boldsymbol{Y} = W_t(\theta)^T\eta + \epsilon^* = \tilde{W}_t(\theta)^T\tilde{\eta} + \epsilon^*, \tag{5.28}$$

where $\tilde{\eta} = D_W\eta$ and D_W is a diagonal matrix consisting of the row norms of $W_t(\theta)$. Hence, by Lemma 1, $\|D_W\tilde{\eta} - D_W\eta\| = \left\|\hat{\tilde{\eta}} - \tilde{\eta}\right\| \leq \frac{4\lambda\sqrt{S}}{1-\delta^W_{2S}-\phi^W_{S,2S}}$ provided (5.26) holds.

Let $\boldsymbol{e}_i$ be a vector of length q with the ith element equals to 1 and all other elements equal to 0. Then $\hat{\beta}_i(t) = (\boldsymbol{e}_i \otimes \boldsymbol{B}_p(t))^T\hat{\eta} = (\boldsymbol{e}_i \otimes \boldsymbol{B}_p(t))^T D_w(\theta)^{-1}\hat{\tilde{\eta}}$ while $\beta_i(t) = (\boldsymbol{e}_i \otimes \boldsymbol{B}_p(t))^T\eta + e_p(t) = (\boldsymbol{e}_i \otimes \boldsymbol{B}_p(t))^T D_w(\theta)^{-1}\tilde{\eta} + e_p(t)$.

Therefore we have

$$\begin{aligned}
|\hat{\beta}_i(t) - \beta_i(t)| &\leq |\hat{\beta}_i(t) - (\boldsymbol{e}_i \otimes \boldsymbol{B}_p(t))^T\eta| + |e_{ip}(t)| \\
&= \left\|(\boldsymbol{e}_i \otimes \boldsymbol{B}_p(t))^T D_w(\theta)^{-1}(\hat{\tilde{\eta}} - \tilde{\eta})\right\| + |e_{ip}(t)| \\
&\leq \left\|(\boldsymbol{e}_i \otimes \boldsymbol{B}_p(t))^T D_w(\theta)^{-1}\right\| \cdot \left\|\hat{\tilde{\eta}} - \tilde{\eta}\right\| + \omega_{ip} \\
&= \frac{1}{\sqrt{nm}}\alpha^{\theta}_{n,p,i}(t)\left\|\hat{\tilde{\eta}} - \tilde{\eta}\right\| + \omega_{ip} \\
&= \frac{1}{\sqrt{nm}}\frac{\alpha^{\theta}_{n,p,i}(t)\lambda\sqrt{S_{pq}}}{1-\delta^W_{2S_{pq}} - \phi^W_{S_{pq},2S_{pq}}} + \omega_{ip}
\end{aligned}$$

5.5 Summary of Results

We have described the model setup for the functional kriging regression model, the algorithm to fit such a model and its good theoretical properties. In this section, we demonstrate the performance of the proposed method by a simulation study and the GM experiment.

5.5.1 Simulation Studies

To evaluate the perfomance of the functional kriging regression model, we first conduct a simulation study based on some known functions. The performance will be evaluated in two aspects: the accuracy of coefficient function estimations and prediction errors. They are both measured by mean integrated square errors calculated based on true coefficient functions and randomly generated testing data. In addition, we will also show the confidence regions for coefficient function estimations. The functional kriging model with the Lasso is illustrated and the results are compared with those according to the regular kriging model. To demonstrate the performance with the increase of the sample size, simulations are conducted for different numbers of design combinations (20 and 30) with 25 and 50 time points for each design setting. The prediction performance is evaluated only for the case of 30 runs.

Four known coefficient functions are defined on the input space $[0, 1]$ and they all belongs to the family of piecewise linear functions but with different magnitudes.

$$\beta_0(t) = \begin{cases} 0, & \text{if } x < 0.4 \\ 0.5t - 0.1 & \text{if } x \geq 0.4 \end{cases} \quad , \quad \beta_1(t) = \begin{cases} 0, & \text{if } x < 0.2 \\ 2t - 0.4 & \text{if } 0.2 \leq x < 0.8 \\ 1.2 & \text{if } x \geq 0.8 \end{cases}$$

$$\beta_2(t) = \begin{cases} t, & \text{if } x < 0.2 \\ 0.2 & \text{if } 0.2 \leq x < 0.8 \\ t - 0.6 & \text{if } x \geq 0.8 \end{cases} \quad , \quad \beta_3(t) = \begin{cases} 1.8 - 3t, & \text{if } x < 0.4 \\ 0 & \text{if } x \geq 0.4 \end{cases}$$

Let $\boldsymbol{\beta}(t) = [\beta_0(t), \beta_1(t), \beta_2(t), \beta_3(t)]^T$ and $\boldsymbol{x} = [1, x_1, x_2, x_3]^T$. Then the response function $y(\boldsymbol{x}, t)$ is defined as:

$$y(\boldsymbol{x}, t) = \boldsymbol{x}^T \boldsymbol{\beta}(t) + Z(\boldsymbol{x}, t)$$

where $Z(x, t)$ is a Gaussian process with mean 0 and the covariance function

$$cov\{Z(x_1, t_1), Z(x_2, t_2)\} = \sigma^2 r(x_1 - x_2, t_1 - t_2),$$

$\boldsymbol{\theta} = (2, 1, 0.5)$ and $\sigma_Z = 0.1$. The response are generated by using the multivariate normal distribution and the experimental designs used are Latin hypercube designs (McKay et al. 1979) with 3 variables and sample sizes $n = 20,\ 30$ and $m = 25,\ 50$. Latin hypercube designs are a popular choice for computer experiments because they are easy to use and fill the design space relatively well. The vector of basis functions, $B(t)$, is chosen to be

$$B(t) = (b_0(t), \ldots, b_{10}(t))^T$$

where $b_0(t) = 1$, $b_1(t) = t$ and $b_i(t) = tI\{t - \xi_{i-1}\}$ for $i = 2, \ldots, 10$ with $\xi_1 = \frac{1}{10}, \xi_2 = \frac{2}{10}, \ldots, \xi_9 = \frac{9}{10}$. For the case of $n = 30$, we will use 25 settings as training data and the remaining 5 settings as testing data.

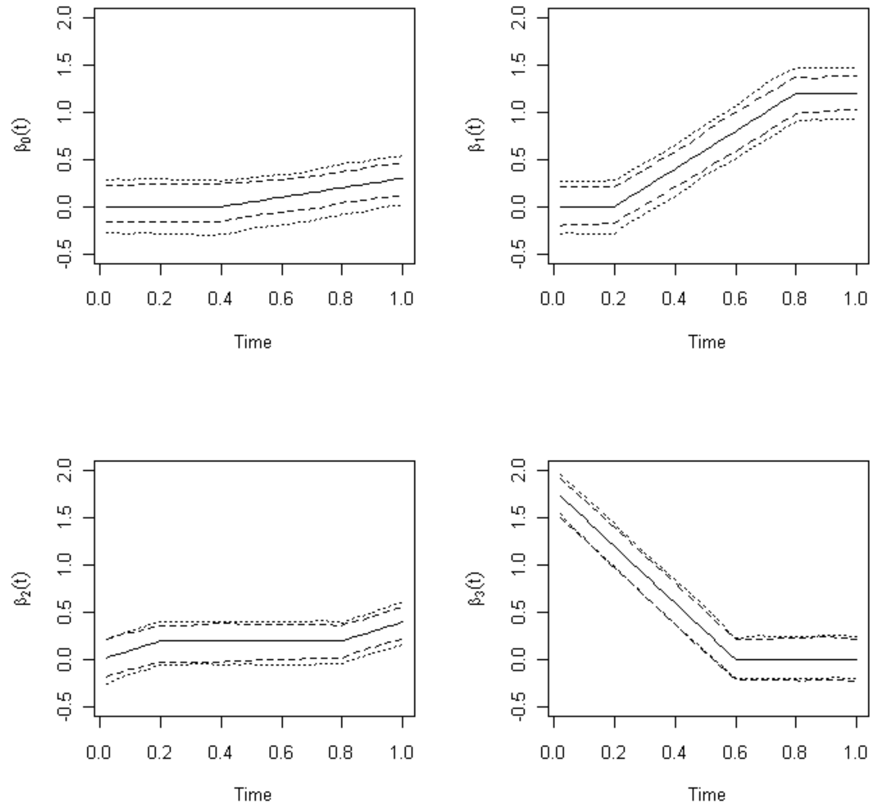


Figure 5.1: A comparison of 90-percent CIs of estimated coefficient functions of unpenalized and penalized kriging models under n = 20. The solid line: True coefficent functions; The dash line: Penalized model; The dotted line: Unpenalized model

Table 5.1: MISE based on known coefficient functions (10^{-3})

Sample Size	X	$n = 20$				$n = 25$			
t		$\beta_0(t)$	$\beta_1(t)$	$\beta_2(t)$	$\beta_3(t)$	$\beta_0(t)$	$\beta_1(t)$	$\beta_2(t)$	$\beta_3(t)$
$m = 25$	Penalized	5.5	3.9	3.9	4.7	2.9	2.9	2.9	2.6
	Unpenalized	9.9	6.6	5.9	4.9	6.4	5.7	5.1	2.8
$m = 50$	Penalized	3.8	3.8	1.9	4.0	3.1	3.1	2.6	3.0
	Unpenalized	8.9	7.8	3.8	4.2	6.7	6.4	4.5	2.9

We compare performance using mean integrated squared error (MISE):

$$\frac{1}{N}\sum_{i=1}^{N}\int_{T}(\hat{y}(t) - y(t))dt/|T|.$$

Based on 100 interations, the performances of coefficient function estimations with penalized and unpenalized functional kriging models are shown in Table 5.1, Figure 5.1 and Figure 5.2. We can see that penalized estimations consistently offer tighter confidence intervals and the benefits of penalized estimation are more significant when the coefficient functions are relatively small. Theorem 1 suggests that the estimation errors become smaller as n and m increase if all other parameters, e.g. δ, ϕ and $C^{\theta}_{n,p,k}(t)$, stay relatively constant. From Table 5.1, we can see that this statement holds true for the number of design settings (n). However, it is not always the case for the number of functional samples (m). This probably suggests that as m increases, either the necessary conditions for Theorem 1 become invalid or δ, ϕ and $C^{\theta}_{n,p,k}(t)$ changes significantly.

Table 5.2: MISE based on 5 test settings

Methods	MISE (10^{-4})	
	$m = 25$	$m = 50$
Penalized	1.19	1.12
Unpenalized	1.30	1.22
PCA-based	4.75	13.4

The MISEs based on the 100 simulations are summarized in Table 5.2 to assess the prediction accuracy of the fitted models. The penalized model performs better

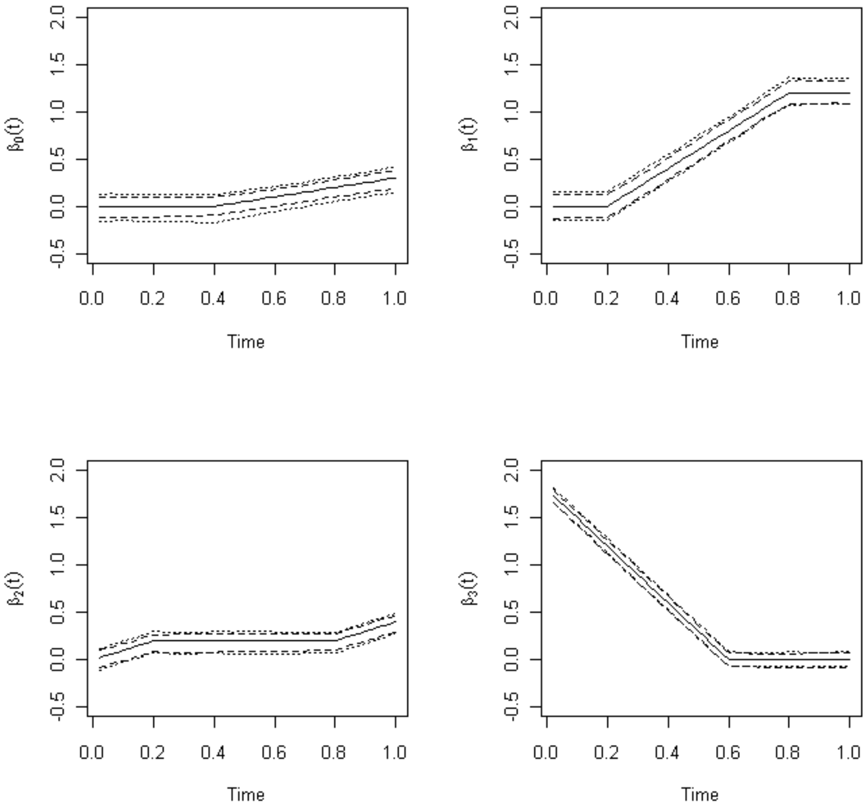


Figure 5.2: A comparison of 90-percent CIs of estimated coefficient functions of unpenalized and penalized kriging models under n = 20. The solid line: True coefficent functions; The dash line: Penalized model; The dotted line: Unpenalized model

than the unpenalized model as expected. And they are both significantly better than the PCA-based kriging model (Dancik et al. (2008)).

5.5.2 GM Experiment

Here we will compare results from fitting the GM acceleration data using the PCA based kriging model with $l = 3$ (Dancik et al. (2008)), the unpenalized functional kriging regression model and the penalized functional kriging regression model. Two types of data are considered: one is the computer experiment output and the other is the discrepancy (bias) between field experiments and computer experiments ($y^b = y^F - y^M$).

For both groups, we will choose 121 data points from all experimental runs. Thus the length of an interval between two consective time points is 1/120 units of rescaled time. As we described in the last section, these 121 data points for each run are interpolated by using existing observations. Piecewise-linear basis functions with 19 knots ($\xi_1 = \frac{1}{20}, \xi_2 = \frac{2}{20}, \ldots, \xi_{19} = \frac{19}{20}$) are used in this study to be consistent with our previous efforts. However, we are considering more knots here to illustrate benefits of the penalized estimation. Under this setting, $B(t)$ is chosen to be

$$B(t) = (b_0(t), \ldots, b_{20}(t))^T$$

where $b_0(t) = 1$, $b_1(t) = t$ and $b_i(t) = tI\{t - \xi_{i-1}\}$ for $i = 2, \ldots, 20$. The design function $f(\boldsymbol{X})$ is assumed to be $(1, x_A, x_B, x_C, x_D, x_{CD} = x_C x_D)^T$ based on premilinary analysis of the data.

Similarly to the study in Chapter IV, we will only consider those 56 "'normal"' runs and randomly divide them into training and testing groups. The training group consists of 25 design combinations (50 runs) and the test group consists of 3 design combinations (6 runs). We first introcuce the estimation results based on the training group and they are followed by the prediction performance of the test group.

5.5.3 Estimation

5.5.3.1 Computor Outputs

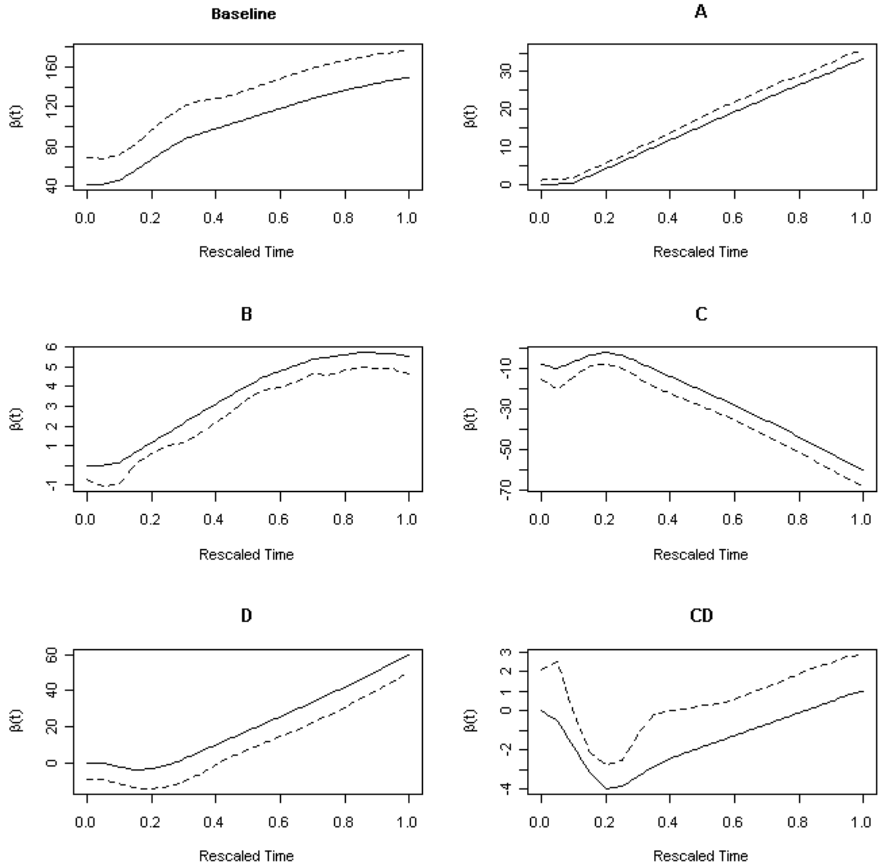


Figure 5.3: A comparison of estimated coefficient functions of unpenalized and penalized kriging models for computer outputs. The solid line: Penalized model; The dash line: Unpenalized model

Figure (5.3) shows the estimated coefficient functions $\beta_0(t)$, $\beta_A(t)$, $\beta_B(t)$, $\beta_C(t)$, $\beta_D(t)$ and $\beta_{CD}(t)$ for unpenalized and penalized kriging models of computer outputs. They present similar shapes for all $f_i(x)$. However, it is clear that the penalized estimation are much smoother than the penalized estimation.

5.5.3.2 Bias Function

Figure (5.4) shows the estimated coefficient functions $\beta_0(t)$, $\beta_A(t)$, $\beta_B(t)$, $\beta_C(t)$, $\beta_D(t)$ and $\beta_{CD}(t)$ for unpenalized and penalized kriging models of bias functions. Unlike

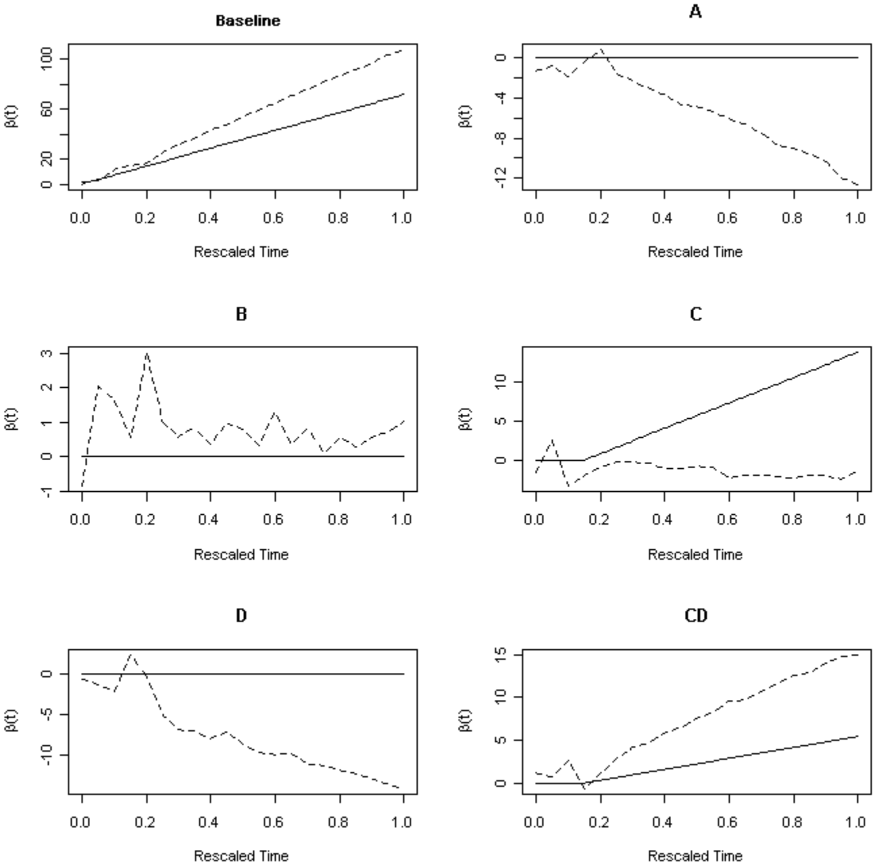


Figure 5.4: A comparison of estimated coefficient functions of unpenalized and penalized kriging models for bias function. The solid line: Penalized model; The dash line: Unpenalized model

results from computer outputs, the estimated coefficient functions differ greatly from each other. The penalized estimations are far more sparse than the regular regression estimations.

5.5.4 Prediction

In addition to the penalized and unpenalized functional kriging models, for comparison of prediction performance, we will also consider the PCA-based kriging model (Dancik et al. (2008)).

5.5.4.1 Computor Outputs

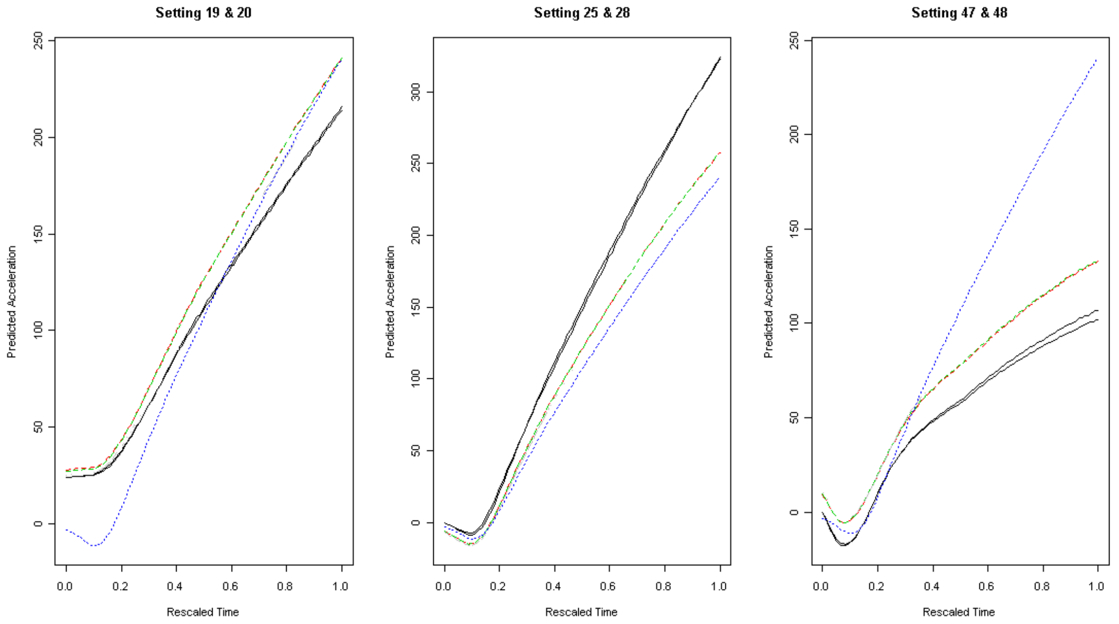


Figure 5.5: A comparison of predictions on the test group. Solid black line: Physical observations; Dashed red line: Penalized model; Dashed green line: Unpenalized model; Dotted blue line: PCA-based Kriging model

The predictions from penalized and unpenalized models show similar performance and they are all better than the PCA-based kriging model. The similarity in performance is due to the fact that the estimated coefficient functions $\beta(t)$'s have similar shapes for both models. It implies that likely true $\beta(t)$'s are not zero for most t in $[0, 1]$. In that case, penalizing the coefficient paramters can't largely improve the prediction. However, it could be proved to much more helpful if we consider a basis system with higher order or more interactions among design variables.

5.5.4.2 Bias between Computer and Physical Outputs

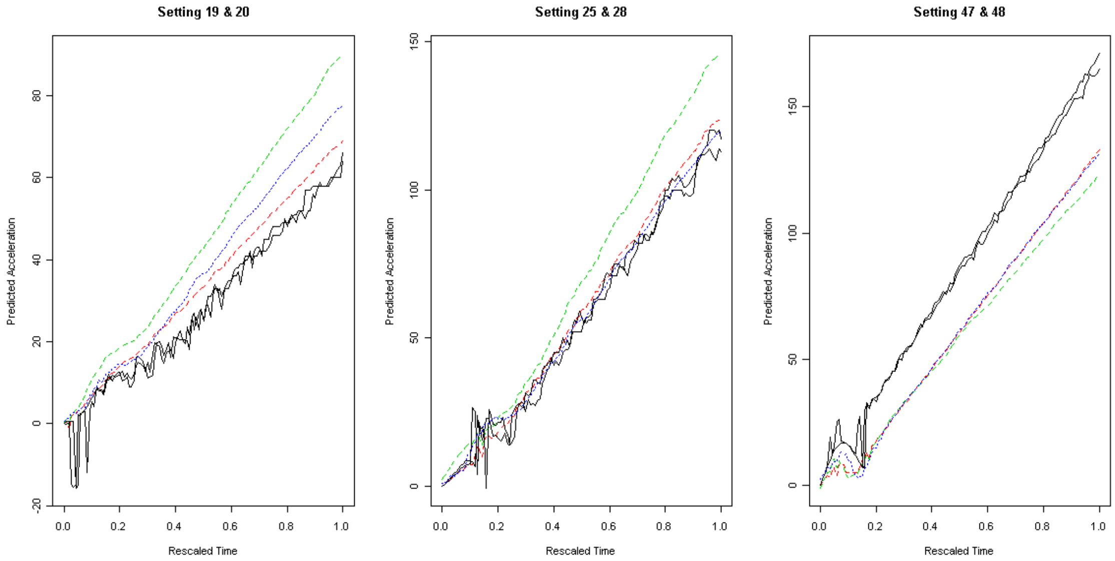


Figure 5.6: A comparison of predictions on the test group. Solid black line: Physical observations; Dashed red line: Penalized model; Dashed green line: Unpenalized model; Dotted blue line: PCA-based Kriging model

In the bias case, we just show that estimated coefficient functions differ significantly between penalized and unpenalized models with penalized estimation showing many zero effects. From Figure 5.6, we can tell that the benefits of additional penalty are more clear in the prediction.

5.6 Discussion

In this work, we develop a penalized functional kriging regression model which achieves model fitting and variable selection simultaneously. This model is a natural extension of our work in Chapter IV and can be considered as a bridge between classical functional data analysis and kriging modeling approaches. The new formulation of the functional kriging model is based on the work of James et al. (2009), which allows the efficient algorithms for estimating penalized parameters. The implementaion also utilizes the results of Hung et al. (2011) to overcome the computational issues involved with correlation matrix operations. The penalized functional kriging model is illustrated by a simulation study and revisiting the same GM experiment in Chapter IV. In simulation study, we show that the proposed model performs better than the one without regularization in terms of coefficient estimation. In GM experiments, we show that in terms of prediction the proposed model performs better than the dimension reduction method for computer outputs and better than both of the dimension reduction method and the kriging model without regularization for bias function. Under certain conditions, we also give tight, non-asymptotic, bounds on the estimation error of the proposed model in Section 5.4.

5.7 Future Research

In this chapter, we only consider the problem with both functional outputs and functional inputs. However, the technique originally developed in James et al. (2009) is targeted at the problem with scalar outputs and functional inputs. It is interesting to see how the proposed method will perform under the orignal problem. In addition, the more general functional linear regression model is given by (Wu et al. 2010)

$$E(Y(t)|X,Z) = \mu_{Y|Z}(t) + \int_{\mathcal{S}} \beta(Z,s,t)(X(s) - \mu_{X|Z}(s))ds. \tag{5.29}$$

It will be also interesting to extend our proposed method to this more general model.

In the functional kriging model, essentially we treat design variables and functional variables in the same way when we construct the covariance matrix. However, it is very common that the shape of the functional curve changes dramaticly from the start point to the end point. Therefore we may want to allow the correlation and variance parameters change over the functional space. However, this flexiblity will require us to develop new efficient algorithm for matrix operations.

Due to the issue of identifiability raised in Loeppky et.al (2006), it is also of interest to investigate more on the relationship between the mean functions and the Gaussian process. By doing so, we may be able to improve the prediction performance of the proposed model.

APPENDIX

Definition of Sprague and Geers Validation Metric for Functional Model

Denote $m(t)$ as the physical outputs, and $c(t)$ is the corresponding computer outputs, then the following time integrals are defined

$$\begin{aligned} \upsilon_{mm} &= (t_2 - t_1)^{-1} \int_{t_1}^{t_2} m^2(t)\mathrm{d}t \\ \upsilon_{cc} &= (t_2 - t_1)^{-1} \int_{t_1}^{t_2} c^2(t)\mathrm{d}t \\ \upsilon_{mc} &= (t_2 - t_1)^{-1} \int_{t_1}^{t_2} m(t)c(t)\mathrm{d}t \end{aligned}$$

where $t_1 < t < t_2$ is the time span of interest for the functional outputs. The error in magnitude is given by

$$M_{SG} = \sqrt{\upsilon_{cc}/\upsilon_{mm}} - 1$$

which is insensitive to phase discrepancies, as it is based upon the area under the squared functional outputs. Equation (9) represents the ratio of the area under the squared computer and physical outputs, with the -1 providing a zero metric value when the two areas are identical.

The phase error by

$$P = \frac{1}{\pi} cos^{-1}(\upsilon_{mc}/\sqrt{\upsilon_{mm}\upsilon_{cc}})$$

which is insensitive to magnitude differences.

The final validation metric is given by

$$C_{SG} = \sqrt{M_{SG}^2 + P^2}$$

which combines magnitude and phase differences.

REFERENCES

[1] American Institute of Aeronautics and Astronautics (1998) *Guide for the verification and validation of computational fluid daynamics simulations.* AIAA-G-077-1998.

[2] Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P. (1996) Experimental design and observation for large systems. *Journal of the Royal Statistical Society, Series B, Methodological*, 58, 77-94.

[3] Bayarri, M. J., Berger, J. O., Higdon, D., Kennedy, M. C., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2007) A Framework for Validation of Computer Models. *Technometrics.* 49(2), 138-154.

[4] Bayarri, M.J., Berger, J.O., Kennedy, M.C., Kottas, A., Paulo, R., Sacks, J., Cafeo, J.A., Lin, C.H. and Tu, J. (2005) Bayesian Validation of a Computer Model for Vehicle Crashworthiness. Technical Report 163, National Institute of Statistical Sciences.

[5] Bayarri, M.J., Berger, J.O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, Parthasarathy, R.J., Paulo, R., Sacks, J., Walsh, D. (2007) Computer model validation with functional output, *Annals of Statistics*, 35(5), 1874-1906.

[6] Campbell, K. (2006) Statistical calibration of computer simulations. *Reliability Engineering and System Safety*, 91(10-11), 1358-1363.

[7] Candes, E. and Tao, T. (2007) The Dantzig selector: Statistical estimation when p is much larger than n (with discussion). *Annals of Statistics* 35(6), 2313-2351.

[8] Chen, W., Xiong, Y., Tsui, K-L., and Wang, S. (2008) Design-Driven Validation Approach using Bayesian Prediction Models, *ASME Journal of Mechanical Design*, 130(2).

[9] Dancik, G.M. and Dorman, K.S. (2008) mlegp: statistical analysis for computer models of biological systems using R, *Bioinformatics*, 24(17), 1966-1967

[10] Dowding, K. J., Pilch, M. and Hills, R. G. (2008) Formulation of the thermal problem, *Computer Methods in Applied Mechanics and Engineering*, 197(29-32), 2385-2389.

[11] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96, 1348-1360.

[12] Fan, J. and Zhang, J. (2000). Two-step estimation of functional linear models with applications to longitudinal data. *Journal of the Royal Statistical Society, Series B* 62, 303-322

[13] Faraway, J. J. (1997). Regression analysis for a functional response, *Technometrics*, 39(3), 254-261.

[14] Ferson, S., Oberkampf, W. L. and Ginzburg, L. (2008). Model validation and predictive capability for the thermal challenge problem, *Computer Methods in Applied Mechanics and Engineering*, 197(29-32), 2408-2430.

[15] Higdon, D., Nakhleh, C., Gattiker, J. and Williams, B. (2008). A Bayesian Calibration Approach to the Thermal Problem, *Comput. Methods Appl. Mech. Engrg.* 197, 2431-2441

[16] Hills, R. G., and Trucano, T. G. (1999). Statistical Validation of Engineering and Scientific Models: Background. SAND99-1256, Sandia National Laboratories, Albuquerque, New Mexico.

[17] Hills, R. G., and Trucano, T. G. (2002). Statistical Validation of Engineering and Scientific Models: A Maximum Likelihood Based Metric. SAND2001-1783, Sandia National Laboratories, Albuquerque, New Mexico.

[18] Hills, R. G., and Trucano, T. G. (2006). Model Validation: Model Parameter and Measurement Uncertainty. *Journal of Heat Transfer*, 128, 339-351.

[19] V. Roshan Joseph, Hung, Y. and Sudjianto, A. (2008). Blind Kriging: A New Method for Developing Metamodels, *ASME (American Society of Mechanical Engineers) Journal of Mechanical Design*, 130, 031102-1-8

[20] Ying Hung (2010). Penalized Blind Kriging in Computer Experiments, *Statistica Sinica*, to appear.

[21] Hung, Y., V. Roshan Joseph and Melkote, S. N. (2009) An Iterative Procedure For Modeling Computer Experiments with Functional Response, Rutgers technical report.

[22] Hung, Y., V. Roshan Joseph and Melkote, S. N. (2011) Analysis of Computer Experiments with Functional Response, submitted.

[23] James, G. M., Wang, J. and Zhu, J. (2009) Functional linear regression that's interpretable, *The Annals of Statistics*, 37(5A), 2083-2108.

[24] James, G. M., Radchenko, P and Lv, J. (2009) DASSO: connections between the Dantzig selector and lasso, *Journal of the Royal Statistical Society, Series B*, 71(1), 127-142.

[25] Kennedy, M. C. and O'Hagan, A. (2000) Predicting the output from a complex computer code when fast approximation are available. *Biometrika*, 87(1), 1-13

[26] Kennedy, M. C. and O'Hagan, A. (2001) Bayesian calibration of computer models (with discussion), *Journal of the Royal Statistical Society B*, 63(3), 425-464

[27] Li, R. and Sudjianto, A. (2005) Analysis of Computer Experiments Using Penalized Likelihood in Gaussian Kriging Models, *Technometrics*, 47, 111-120.

[28] Liu, F., Bayarri, M.J., Berger, J., Paulo, R., Sacks, J. (2008) A Bayesian analysis of the thermal challenge problem, *Comput. Method Appl. Mech.*, 197, 2457-2466

[29] Loeppky, L.J., Bingham, D., Welch, W.J. (2006) Computer Model Calibration or Tuning in Practice, Technometrics, under revision.

[30] Mardia, K. V. and Marshall, R. J. (1984) Maximum likelihood estimation of models for residual covariance in spatial regression , *Biometrika*, 71(1), 135-146

[31] McFarland, J. *UNCERTAINTY ANALYSIS FOR COMPUTER SIMULATIONS THROUGH VALIDATION AND CALIBRATION*, PhD thesis, Vanderbilt University, 2008.

[32] McFarland, J., Mahadevan, S., Romero, V. and Swiler, L. (2008) Calibration and Uncertainty Analysis for Computer Simulations with Multivariate Output, *AIAA Journal*, 46(5), 1253-1265

[33] Oberkampf, W. L. and Barone, M. F. (2006) Measures of Agreement Between Computation and Experiment: Validation Metrics. *Journal of Computational Physics*, 217, 5-36.

[34] Nair, V. N., Taam, W. and Ye, K. Q. (2002) Analysis of Functional Responses From Robust Design Studies, Journal of Quality Techonology, 34(4), 355-370.

[35] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989) Design and Analysis of Computer Experiments. *Statistical Science*, 4(4), 409-435.

[36] Saltelli, A., Chan, K. and Scott, E. (2000) *Sensitivity Analysis*. John Wiley & Sons, Chichester.

[37] Santner, T. J., Williams, B. J., and Notz, W. I. (2003) *The Design and Analysis of Computer Experiments*. Springer.

[38] Ramsay, J. O. and Silverman, B.W. (2002) Functional Data Analysis. 2nd Ed. Springer-Verlag.

[39] Schwer, L. E. (2007) Validation Metrics for response histories: perspectives and case studies, *Engineering with Computers*, 23, 295-309

[40] Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* 58, 267-288.

[41] Trucano, T.G., Swiler, L.P., Igusa, T., Oberkampf, W.L. and Pilch, M. (2006) Calibration, validation, and sensitivity analysis: What's What, *Reliability Engineering and System Safty*. 91 (10-11), pp. 1331-1357

[42] Wang, S., Chen, W. and Tsui, K-L. (2009) Bayesian Validation of Computer Models, *Technometrics*, in revision.

[43] 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(1), 15-25.

[44] Xiong, Y., Chen, W. and Tsui, K-L. (2009) A better understanding of model updating strategies in validating engineering models. *Computer Methods in Applied Mechanics and Engineering*, 198(15-16), 1327-1337 .

[45] Zhou, L., Huang, J., Martinez, J.G., Maity, A., Baladandayuthapani, V. and Carroll, R. J. (2010) Reduced Rank Mixed Effects Models for Spatially Correlated Hierarchical Functional Data. *Journal of the American Statistical Association*, 105(489): 390-400.

[46] Zou, H. and Hastie, T. (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B* 67, 301-320.

VITA

Xuyuan Liu was born on the 29st of June 1983 in Tianjin, China. He is the only child of his parents. After attending Yaohua High School, Xuyuan attended Peking University in Beijing, China earning a Bachelor's degree in Statistics in 2004. After the undergraduate study, he went abroad to attend University of Waterloo in Waterloo, ON earning a Master's degree in Statistics in 2006 under the supervision of Professor Stefan Steiner. In August of 2006 he joined the school of Industrial and Systems Engineering at Georgia Tech to pursue a Ph.D. in Applied Statistics under the supervision of Professor Kwok-Leung Tsui. In his free time, Xuyuan enjoy playing soccer and badminton and reading historical novels.