MODELING THE OUTPUT FROM COMPUTER EXPERIMENTS HAVING QUANTITATIVE AND QUALITATIVE INPUT VARIABLES AND ITS APPLICATIONS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

Gang Han, M.S.

* * * * *

The Ohio State University

2008

Dissertation Committee:

Dr. Thomas J. Santner, Adviser

Dr. William I. Notz

Dr. Angela M. Dean

Approved by

Adviser

Graduate Program in Statistics

ABSTRACT

Complex computer models have extensive usage in scientific and engineering studies. Because the number of computer runs is typically limited, statistical models are used to predict the computer codes. This thesis considers two research problems. The first problem is prediction for computer experiments having quantitative and qualitative mixed input variables. The second is the simultaneous determination of tuning and calibration parameters.

To predict the output from a computer experiment having mixed inputs, we regard the output from a computer experiment code as a realization from a mixture of Gaussian Stochastic Processes (GaSPs) and have developed two methods. The first method assumes that the responses at different qualitative input levels share similarities. We build one GaSP model for each level of the qualitative input. Using Bayesian hierarchical models with an empirical prior, the predictions at one qualitative input level are able to borrow information from the responses at other levels. The second method estimates the common trend of the responses at all the qualitative input levels. The prediction is the sum of the estimated average and the predicted deviation of a response from the average. We develop a data adaptive algorithm for the estimation of the common trend to guarantee that the predictive error of this predictor is no bigger than that of a predictor using the data at one level only. We extend the both methods to computer experiments having multiple qualitative inputs.

To simultaneously select tuning and calibration parameters, we develop a Bayesian discrepancy-based procedure to estimate the tuning parameters and simulate the estimated posterior distribution of the calibration parameters.

We compare our methodologies with alternatives and implement the methodologies in three biomechanical engineering applications.

To my parents.

ACKNOWLEDGMENTS

It has been said that any one should have at least three great supporters to succeed. The completeness of my Ph.D. study is impossible without the support from many faculty members and graduate students at the Ohio State University.

First of all, I shall express my gratitude to professor Thomas Santner, my Ph.D. thesis adviser. Dr. Santner not only guided my research, but also taught me to think and behave as a professional statistician. I am impressed by his deep and broad knowledge and I am indebted to his patient guidance. I also thank Gail, Dr. Santner's wife, for the delicious food and the warm-hearted conversations.

I would like to thank many professors in the Department of Statistics at the Ohio State University. I am grateful to Dr. William Notz, who led my way to the research of computer experiment at 2004 and guided me in the following 5 years. I thank Dr. Elizabeth Stasny, the graduate student chair, for her warm-hearted encouragement and support. I would like to thank Drs. Angela Dean, Peter Craigmile, Tao Shi, and Greg Allenby for being in my thesis defense committee and candidate exam committee and providing me insightful comments.

This thesis would not be finished without the support from Cornell University and the Hospital for Special Surgery. I would like to give special thanks to Donald Bartel, Jason Long, and Jeremy Rawlinson for their collaboration and constructive

suggestions. I also want to thank Brian Williams for suggesting the use of a shrinkage prior in Chapter 2.

I would like to thank all the students joining in the department in 2003. During the times exams and thesis research, we worked and cheered together. Thank you for the support. Thanks for the sweet and colorful memories!

Finally, I am indebted to my parents, Han, Yusheng and Liu, Shufang, for their everlasting love. This thesis is dedicated to them.

VITA

December, 1980	Born - Beijing, China
2003	B.S. Computer Science, The Beijing University of Technology
2005	M.S. Statistics, The Ohio State University
2006	Intern, Statistical and Applied Mathematical Sciences Institute
2005-2008	Graduate Research Associate and Teaching Associate, The Ohio State University.

PUBLICATIONS

Research Publications

Han, G., Santner, T. J., Notz, W. I., and Bartel, D. L., "Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables." *Accepted, Technometrics*, 2008.

Han, G., Santner T. J., and Rawlinson J. J., "Simultaneous Determination of Calibration and Tuning Parameters," *Submitted, Technometrics*, 2008.

FIELDS OF STUDY

Major Field: Statistics
Studies in: The design and analysis of complex computer models

TABLE OF CONTENTS

Page

Abstract ii

Dedication iv

Acknowledgments v

Vita vii

List of Tables xii

List of Figures xiii

Chapters:

1. Introduction 1

    1.1 An Overview of Computer Experiments 2
        1.1.1 Physical Experiments – the Gold Standard 2
        1.1.2 Computer Experiments 3
        1.1.3 A Motivating Example 3
    1.2 Inputs to Computer Experiments 4
        1.2.1 Quantitative and Qualitative Inputs 4
        1.2.2 Control Variables and Calibration/Tuning Parameters . . . 5
    1.3 The Design of Computer Experiments 6
    1.4 Gaussian Stochastic Process Models 8
        1.4.1 Introduction 8
        1.4.2 Inferences about the Model Parameters 9
        1.4.3 Prediction 10
    1.5 The Metropolis-Hastings (M-H) Sampling Algorithm 12

2. Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables 14

2.1 Introduction 14
2.2 Prediction for Computer Experiments Having Quantitative Input(s) and One Qualitative Input 15
2.3 Prediction for Computer Experiments Having Quantitative Input(s) and Multiple Qualitative Inputs 22
2.4 Comparing the HQQV Predictor with Three Competing Predictors 27
2.4.1 Competing Predictors 27
2.4.2 Comparison of $\widehat{y}^{SHB}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ 30
2.4.3 Interpolation and Extrapolation Accuracies of $\widehat{y}^{SHB}(\cdot)$, $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{PBP}(\cdot)$, and $\widehat{y}^{HQV}(\cdot)$ 34
2.5 An Application of the Multivariate HQQV Model in Biomechanical Engineering 38
2.6 Summary and Future Research 40

3. ANOVA Kriging: A Methodology for Predicting the Output from a Complex Computer Code Having Quantitative and Qualitative Inputs 43

3.1 Introduction 43
3.2 The ANOVA Kriging Model for Computer Experiments Having One Qualitative Input 45
3.2.1 Ordinary Kriging and Average Effect Kriging 45
3.2.2 Comparing $\widehat{y}^{OK}(\cdot)$ and $\widehat{y}^{AE}(\cdot)$ 48
3.2.3 The ANOVA Kriging Predictor for Computer Experiments Having an Arbitrary Number of Quantitative Inputs and One Qualitative Input 52
3.2.4 The HAK Predictor 53
3.2.5 An Example Having One Quantitative Input and One Qualitative Input Having Three Levels 54
3.3 The HAK Predictor for Computer Experiments Having Multiple Qualitative Inputs 56
3.3.1 The HAK Model for Computer Experiments Having Two Qualitative Inputs 56
3.3.2 The HAK Predictor for Computer Experiments Having an Arbitrary Number of Qualitative Inputs 58
3.3.3 An Example Having One Quantitative Input and Two Qualitative Inputs 59
3.4 An Application of the HAK Predictor to a Hip Resurfacing System 61
3.5 Summary and Future Research 67

4. Simultaneous Calibration and Tuning for computer experiments 69

4.1 Introduction 69
4.2 A Hierarchical Bayesian Model for Tuning and Calibration 74
4.3 Methodology for Simultaneous Tuning and Calibration 79
4.3.1 The Discrepancy Function 79
4.3.2 Simultaneous Tuning and Calibration 80
4.3.3 Prediction 83
4.4 Examples and Comparison 85
4.4.1 Discussion 85
4.4.2 An Illustrative Example with Known $\boldsymbol{t}^\star$ and θ_c 87
4.4.3 A Biomechanics Example 91
4.5 Summary and Future Research 93

Bibliography 94

LIST OF TABLES

Table **Page**

3.1 True responses of the testing data inputs and predictions of $\widehat{y}^{HAK}$, $\widehat{y}^{HAK}_{1,2,3}$, and $\widehat{y}^{HAK}_{1,2,3,12}$. 66

4.1 Specifications of the Metropolis-Hastings algorithm. The four columns, from left to right, correspond to the prior distributions, the lower and upper bounds of the parameters as the program iterates, the initial values of the parameters, and the lengths of the uniform distributions. We let $\mathrm{TN}(\mu, \sigma^2)$ on $[a, b]$ denote the truncated normal distribution with mean μ and variance σ^2 on the support $[a, b]$. 79

4.2 Grid of t and the approximate integral (4.14) 90

LIST OF FIGURES

Figure **Page**

2.1 A simulated surface with $\rho = 0.5$ (the left panel) and a simulated surface with $\rho = 0.9$ (the right panel). 31

2.2 Four plots of the RMSPE comparisons of $\widehat{y}^{SHB}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ for 100 test surfaces. The upper left panel uses $(\rho, n) = (0.5, 8)$, the upper right panel uses $(\rho, n) = (0.5, 20)$, the lower left panel uses $(\rho, n) = (0.9, 8)$, and the lower right panel uses $(\rho, n) = (0.9, 20)$. In each panel, the horizontal axis corresponds to the RMSPE of $\widehat{y}^{SHB}(\cdot)$; the vertical axis corresponds to the RMSPE of $\widehat{y}^{HQV}(\cdot)$; the solid line is the 45 degree line passing through the origin; the circles denote the RMSPEs of $\widehat{y}^{HQV}(\cdot)$ against $\widehat{y}^{SHB}(\cdot)$; a circle below the 45 degree line indicates that $\widehat{y}^{HQV}(\cdot)$ has a smaller RMSPE than $\widehat{y}^{SHB}(\cdot)$. 33

2.3 Plots of the true responses (solid curves) and the training data (circles) for one draw using Process 1 (the left panel), Process 2 (the middle panel), and Process 3 (the right panel). 35

2.4 Boxplots of the interpolation RMSPEs of $\widehat{y}^{SHB}(\cdot)$, $\widehat{y}^{PBP}(\cdot)$, $\widehat{y}^{KOH}(\cdot)$, and $\widehat{y}^{HQV}(\cdot)$. The three panels correspond to Process 1 (the left panel), Process 2 (the middle panel), and Process 3 (the right panel). 36

2.5 Boxplots of the 30 extrapolation RMSPEs of $\widehat{y}^{SHB}(\cdot)$, $\widehat{y}^{PBP}(\cdot)$, $\widehat{y}^{KOH}(\cdot)$, and $\widehat{y}^{HQV}(\cdot)$. The three panels correspond to Process 1 (the left panel), Process 2 (the middle panel), and Process 3 (the right panel). 36

2.6 Predicted APD surfaces for the four combinations of (Prosthesis Design, Loading Pattern). The four combinations are (a) *(CR,NG)*: the upper left panel, (b) *(CR,SC)*: the upper right panel, (c) *(PS,NG)*: the lower left panel, and (d) *(PS,SC)*: the lower right panel. The two quantitative inputs are the Initial Position and the Interface Friction. 40

3.1 Boxplots of the 30 interpolation RMSPEs of $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{HQV}(\cdot)$, and $\widehat{y}^{HAK}(\cdot)$. The three panels correspond to Mechanism 1 (the left panel), Mechanism 2 (the middle panel), and Mechanism 3 (the right panel). 55

3.2 Boxplots of the 30 extrapolation RMSPEs of $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{HQV}(\cdot)$, and $\widehat{y}^{HAK}(\cdot)$. The three panels correspond to Mechanism 1 (the left panel), Mechanism 2 (the middle panel), and Mechanism 3 (the right panel). 55

3.3 Raw data and the four true curves. Observations on the curves at level $(1,1)$, $(2,1)$, $(1,2)$, and $(2,2)$ are denoted by plus signs, circles, triangles, and squares, respectively. 60

3.4 The true response $y(1,1,\cdot)$ (the solid curve), $\widehat{y}^{HQV}(1,1,\cdot)$ (the dashed dots), $\widehat{y}_1^{HAK}(1,1,\cdot)$ (the solid points), and $\widehat{y}_{1,2}^{HAK}(1,1,\cdot)$ (the x-mark). 61

3.5 Two bones and the finite element models of the two bones with implants. Upper left panel: Bone 1; upper right panel: Bone 2; lower left panel: the finite element model of Bone 1 with its implant; lower right panel: the finite element model of Bone 2 with its implant. Source: Long (2008). 63

4.1 Scatter plot of the measured APD over the gait cycle, from the knee simulator (triangles) and the FEA computer code (dots). 72

4.2 Simulated posterior distributions of load discretization (the left panel) and initial position (the right panel). 73

4.3 Simulated posterior distributions of θ_t (the left panel), θ_{c_1} (the middle panel), and θ_{c_2} (the right panel) using the Bayesian calibration program. 88

4.4 The training data (solid circles), the true response curve (the solid line), and the predictions (pluses) obtained by `gpmsa`. 89

4.5 Simulated posterior distributions of θ_{c_1} (the left panel) and θ_{c_2} (the right panel) using STaC 90

4.6 The training data (solid circles), true response curve (the solid line), predictions (pluses), and 99% prediction bands (dashes) using the STaC program. 91

4.7 A plot of the estimated squared discrepancy against the value of the tuning parameter (the left panel) and a histogram of the simulated posterior distribution of the calibration parameter (the right panel). . 92

4.8 The training data (triangles), predictions (pluses), and 99% prediction bands (dashes) using the STaC program. 92

CHAPTER 1

INTRODUCTION

This chapter introduces concepts, models, and numerical methods that will be used later. Section 1.1 introduces physical experiments, mechanical simulators, and computer experiments. Section 1.2 defines different types of the input to computer experiments. Section 1.3 sketches the computer experimental designs that will be used in the later chapters. Section 1.4 describes the Gaussian stochastic process model. Section 1.5 discusses the Metropolis-Hastings algorithm for simulating the posterior distribution.

The rest of the thesis is organized as follows. Chapters 2 and 3 propose a hierarchical Bayesian method and an ANOVA method, respectively, for the prediction for computer experiments having quantitative and qualitative inputs. Chapter 4 proposes a discrepancy-based methodology together with a Bayesian model for the simultaneous tuning and calibration.

1.1 An Overview of Computer Experiments

1.1.1 Physical Experiments – the Gold Standard

In a physical experiment, one measures a stochastic response with a set of treatment input variables (Dean and Voss (1999)). Because of the random error and recognized and unrecognized nuisance parameters, *randomization*, *blocking*, and *replication* are useful techniques to improve the experimental validity. Randomization helps to prevent the confounding between the unrecognized nuisance parameters and the treatment factors. Blocking removes the effect of the recognized nuisance parameters. Replication is needed to estimate the random error.

Physical experiments are the mainstay of agriculture, industry, and medical research. However, physical experiments can be hard or impossible to run. For example, physical experiments studying climate changes, the efficacy of prosthetic devices, and cosmic phenomenon are hard or impossible to conduct. (See Santner, Williams and Notz (2003), chapter 1 and Fang, Li and Sudjianto (2005), chapter 1 for details.) For such a physical experiment, mechanical simulator and computer simulation code can be used to approximate the response.

One type of physical experiment used in biomechanics is based on mechanical simulator of the physical phenomenon. In some of the computer experiments we consider, the computer codes are used to mimic the mechanical simulator. In Chapter 4, we will analyze the output from a computer experiment and that from a mechanical simulator.

1.1.2 Computer Experiments

Computer simulations that implement the mathematical models describing the input-output relationships in the physical experiments are used as surrogates when physical experiments are difficult or impossible to conduct. In a computer experiment, a (complex) numerical code relates the important inputs to the outputs of engineering or scientific interest. Computer codes are also used to supplement physical experiments. Computer codes that serve as the basis for computer experiments have running times that can range from minutes to days (Santner et al. (2003), chapter 1). Thus, statistical predictive models are typically needed to infer the responses at input points that have not been run.

1.1.3 A Motivating Example

We introduce a biomechanical engineering application having a mechanical simulator and a complex computer simulation code. Rawlinson, Furman, Li, Wright and Bartel (2006) compared the Install-Burstein (IB) knee implant produced by Zimmer, Inc. and the Optetrak knee implant produced by Exactech, Inc using finite element code and a knee testing machine. In this application, a hypothetical physical experiment would relate in vivo damage with the implant properties and the patient conditions. Such experiments are not performed.

An Instron-Stanmore KC1 loaded control knee simulator, which is produced by Instron Engineering Corporation, related the damage to kinematics. However, the knee simulator was unable to measure the kinetics and the stresses, which were believed to cause damage to the knee implants. A finite element analysis (FEA) computer code

simulated the kinematics, kinetics, and stresses of the two implants. The FEA code provided useful information for reducing the overall forces in the knee implants.

1.2 Inputs to Computer Experiments

We define different types of inputs to computer experiments. Section 1.2.1 illustrates quantitative and qualitative inputs. Section 1.2.2 introduces control variables, tuning parameters, and calibration parameters.

1.2.1 Quantitative and Qualitative Inputs

Some computer codes have only quantitative inputs. However, many computer codes have both quantitative inputs as well as inputs that are nominal valued. For example, Rawlinson et al. (2006) implemented a finite-element analysis to determine the kinematics and kinetics of prosthetic joints. Their computer codes included numerous quantitative inputs such as the prosthesis material-properties and patient bone material-properties. Their computer codes also included nominal valued qualitative inputs such as "the knee-loading configuration." which could be set to either "gait walking" or "stair climbing."

Another example of a computer experiment having both quantitative and qualitative inputs is described in Qian and Wu (2006) where a computer code determines several room air characteristics of a data storage area. The quantitative inputs included the measurements of the room such as the volume and the height. The qualitative inputs in the computer code were the location of an air diffuser unit, the location of a hot-air return vent, and the type of power unit used.

Whereas many statistical models have been developed for prediction when a computer experiment has only quantitative inputs, relatively few efforts have been dedicated to the statistical modeling of computer experiments having quantitative and qualitative mixed inputs. In Chapter 2 and 3, we will propose two predictors to address this issue.

1.2.2 Control Variables and Calibration/Tuning Parameters

Control variables (such as engineering design inputs) are inputs that are controllable in both the computer code and the corresponding physical experiment (Santner et al. (2003)). Tuning parameters and calibration parameters are controllable in the running of the computer code but not the physical experiment.

Tuning parameters are typically numerical quantities that control the solution of a numerical algorithm implemented in a computer experiment. Tuning parameters have no meaning in the physical experiment. For example, Cox, Park and Singer (1996) studied a computer code simulating the time scale over which nuclear energy could leak out of the plasma in a tokamak nuclear fusion reactor. The tuning parameter was a certain coefficient in the mathematical equation implemented by this code. Thus, tuning is the process of determining the values of the tuning parameters so that a computer simulation can best represent the corresponding physical experiment.

On the other hand, calibration parameters have meanings in the physical experiment but are either unknown or unmeasured during the running of the physical experiment. For example, Kennedy and O'Hagan (2001) described a computer code for simulating the deposition of ruthenium 106 in the Tomsk-7 chemical plant that

caused an accident in 1993. In their code, one calibration parameter was the deposition velocity. Calibration is the process of determining plausible values of the calibration parameters using a limited number of observations from the computer and the physical experiments.

Some computer experiments have both tuning and calibration parameters. For example, Rawlinson et al. (2006) described a Finite Element Analysis (FEA) computer code simulating the forces and movements of a knee prosthesis under a given loading regimen. The two tuning parameters were mesh density and load discretization used to describe the knee loading. The two calibration parameters were friction between the bone and the prosthesis and the position of the femur relative to the tibial tray in the initial gait cycle.

Although considerable research has been dedicated to setting tuning or calibration parameters, there has been little effort on setting both simultaneously. We introduce a methodology for the simultaneous determination of tuning and calibration parameters in Chapter 4.

1.3 The Design of Computer Experiments

Because the number of runs in a computer experiment is often limited, a reasonable design of the input points is necessary in the study of computer experiments. A design should be based on the research objective. For example, to predict the output from computer experiments, a design that helps improve the predictive accuracy is favorable, while for optimization purposes, a (sequential) design that can correctly find the minimum (maximum) is desirable.

In this thesis, the ideal design is one that can best help explore the response surface of the computer code output (or the response from a physical experiment). Thus, we choose to use space-filling designs whose goal is to evenly fill up the design space with input points (Santner et al. (2003), chapter 5). Specifically, we implement the Maximin Latin Hypercube Design (MmLHD) in our examples because MmLHD has clear intuition and is easy to generate (Johnson, Moore and Ylvisaker (1990)). We briefly introduce the Latin Hypercube Design (LHD) and the MmLHD next.

The design matrix of an LHD with n runs and d inputs, which is denoted as LHD(n, d), is an $n \times d$ matrix each of whose columns is a permutation of the integers $\{1, 2, \ldots, n\}$. Latin hypercube design was first developed for numerical integration. McKay, Beckman and Conover (1979) proved that the sample mean of an LHD converges to the population mean almost everywhere and that the variance of the sample mean of an LHD is smaller than the variance of the sample mean of a simple random sampling under mild conditions on the function being evaluated. Further, it is obvious that the projection of an LHD to each of the d inputs has n points evenly spread. However, generic LHDs may behave poorly in terms of the space-filling property and thus may result in poor estimation of the model parameters and biased prediction of unknown responses. This is because an LHD can be regarded as a specific form of the stratified sampling, which does not involve any criterion measuring the space-filling properties. For example, when each column of the design matrix is $(1, 2, \ldots, n)^\top$, the design points will lie on a line in the d dimensional space, which does not seem to be space-filling. Latin Hypercube designs have therefore been integrated with other criteria or other types of designs such as the Maximin criterion (Johnson et al. (1990)) and orthogonal arrays (Tang (1993) and Wu and Hamada (2000)). The MmLHD

having n points in d dimensions is an LHD that maximizes the minimum distance between all pairs of the design points among all the possible LHD(n, d) designs. Thus in this article, our $n \times d$ design is an LHD design with the property that the smallest distance between any two design points are maximized.

1.4 Gaussian Stochastic Process Models

1.4.1 Introduction

Gaussian **S**tochastic **P**rocesses (GaSPs) are used ubiquitously for modeling outputs of computer experiments (Sacks, Welch, Mitchell and Wynn (1989b), Sacks, Schiller and Welch (1989a), Currin, Mitchell, Morris and Ylvisaker (1991), Morris, Mitchell and Ylvisaker (1993), and Santner et al. (2003). Their flexibility, tractability, and interpolating properties make them generally the most popular models for the study of computer experiments. This section provides a brief overview of GaSP models. The models described here are closely related to the models that we develop in Chapter 2, 3, and 4.

We let $y(\cdot)$ denote the response from a computer experiment and $\boldsymbol{x} \in [0,1]^d$ (or can be so scaled) denote an input having d components. The GaSP model views $y(\cdot)$ as a realization from stochastic process

$$Y(\boldsymbol{x}) = f^\top(\boldsymbol{x})\boldsymbol{\beta} + Z(\boldsymbol{x}), \tag{1.1}$$

where $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_q(\boldsymbol{x}))^\top$ is a $q \times 1$ vector with the elements being real-valued functions of inputs $\boldsymbol{x}$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q)^\top$ is a $q \times 1$ vector of unknown regression parameters. The process $Z(\cdot)$ is a stationary Gaussian process with mean 0 and covariance between two responses $Z(\boldsymbol{x}_1)$ and $Z(\boldsymbol{x}_2)$

$$Cov(Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)) = \sigma_Z^2 R(\boldsymbol{x}_1 - \boldsymbol{x}_2), \tag{1.2}$$

where σ_Z^2 denotes the process variance and $R(\boldsymbol{x}_1 - \boldsymbol{x}_2)$ denotes the correlation function of $Z(\cdot)$.

Specifically, we consider *the product exponential correlation function*; i.e.,

$$R(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \prod_{i=1}^{d} \rho_i^{|x_{1,i} - x_{2,i}|^{\alpha_i}}, \tag{1.3}$$

where $0 \leq \theta_i \leq 1$ and $0 < \alpha_i \leq 2$ denote the correlation parameters and $x_{1,i}$ and $x_{2,i}$ denote the ith components of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for all $i = 1, \ldots, d$. The ρ_i's indicate the degree to which the correlation decreases as the distance between the ith dimension of $\boldsymbol{x}_1$ and that of $\boldsymbol{x}_2$ grows. When ρ_i decreases, $Cov(\boldsymbol{x}_1, \boldsymbol{x}_2)$ approaches 0 faster for fixed $|x_{1,i} - x_{2,i}|$. When ρ_i is 0, the $Y(\boldsymbol{x}_1)$ and $Y(\boldsymbol{x}_2)$ become independent if $x_{1,i} \neq x_{2,i}$. The parameters $\alpha_1, \ldots, \alpha_d$ are referred to as *the smoothness parameters* controlling the roughness of the random function $Y(\cdot)$. Specifically, $y(\cdot)$ becomes rougher as α_i approaches 0. In this thesis, I build GaSP models with the product exponential correlation function having the smoothness parameters equal to 2. This correlation function is also known as *Gaussian correlation.*

1.4.2 Inferences about the Model Parameters

There are two methodologies for estimating the model parameters. The first is the frequentist methodology, which estimates the GaSP model parameters. Some estimators proposed in the literature are the **M**aximum **L**ikelihood **E**stimator (MLE), **RE**stricted **M**aximum **L**ikelihood estimator (REML), and the cross validated estimator. (See Santner et al. (2003), page 65 – 68, for details about frequentist estimators.)

The second methodology is Bayesian. The idea of the Bayesian estimation is to propose a prior distribution for each of the model parameters and then study the corresponding posterior distribution. Specifically, let $\boldsymbol{\phi} = (\beta, \sigma_Z^2, \boldsymbol{\rho})$ denote all the

model parameters, $[\boldsymbol{\phi}]$ denote the prior, and $\boldsymbol{y}^s = (y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_n))^\top$ denote the runs from the computer simulation. The posterior distribution of $\boldsymbol{\phi}$ is proportional to the product of the prior density and the likelihood; i.e.,

$$[\boldsymbol{\phi}|\boldsymbol{y}^s] \propto [\boldsymbol{\phi}] \times [\boldsymbol{y}^s|\boldsymbol{\phi}]. \tag{1.4}$$

In Chapter 2, 3, and 4, we simulate the posterior distribution of the parameters in the proposed model following (1.4).

Based on the (simulated) posterior distribution of $\boldsymbol{\phi}$, a point estimator of $\boldsymbol{\phi}$ can be obtained as the posterior mean, median, or mode, depending on one's loss function. The uncertainty in this point estimator can be evaluated by the variance of the posterior distribution. In this paper, we will study the simulated posterior distributions to make inferences about certain parameters in the model and to predict unknown responses. We introduce the frequentist and the Bayesian predictors for an unknown observation $y(\boldsymbol{x}_0)$ next.

1.4.3 Prediction

The basis of both the frequentist and the Bayesian predictor is the conditional normal distribution. Given parameters $\boldsymbol{\phi} = (\boldsymbol{\beta}, \sigma_Z^2, \boldsymbol{\rho})$, $y(\boldsymbol{x}_0)$ and $\boldsymbol{y}^s$ are realizations from a random function $Y(\boldsymbol{x}_0)$ and a random vector $\boldsymbol{Y}^s$, which jointly distributed as

$$\begin{bmatrix} Y(\boldsymbol{x}_0) \\ \boldsymbol{Y}^s \end{bmatrix} \sim MVN\left(\begin{pmatrix} \boldsymbol{f}^\top(\boldsymbol{x}_0) \\ \boldsymbol{F}^\top \end{pmatrix} \boldsymbol{\beta}, \sigma_Z^2 \begin{pmatrix} 1 & \boldsymbol{r}_0^\top \\ \boldsymbol{r} & \boldsymbol{R} \end{pmatrix}\right), \tag{1.5}$$

where $\boldsymbol{F}$ is an $q \times n$ matrix whose jth column is $\boldsymbol{f}(\boldsymbol{x}_j)$, $\boldsymbol{r}_0$ is an $n \times 1$ vector whose jth element is $R(\boldsymbol{x}_0 - \boldsymbol{x}_j)$, and R is an $n \times n$ matrix whose (j,k)th element is $R(\boldsymbol{x}_j - \boldsymbol{x}_k)$. Thus, given the training data $\boldsymbol{y}^s$ and parameters $\boldsymbol{\phi}$, $Y(\boldsymbol{x}_0)$ has the conditional

distribution

$$[Y(\boldsymbol{x}_0)|\boldsymbol{y}^s, \boldsymbol{\phi}] \sim N(\boldsymbol{f}^\top(\boldsymbol{x}_0)\boldsymbol{\beta} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^s - \boldsymbol{F}^\top\boldsymbol{\beta}), \sigma_Z^2(1 - \boldsymbol{r}_0^t \boldsymbol{R}_0^{-1}\boldsymbol{r}_0)). \tag{1.6}$$

The conditional mean $\boldsymbol{f}^\top(\boldsymbol{x}_0)\boldsymbol{\beta} + \boldsymbol{r}_0^\top \boldsymbol{R}^{-1}(\boldsymbol{y}^s - \boldsymbol{F}^\top\boldsymbol{\beta})$ is the **B**est **L**inear **U**nbiased **P**redictor (BLUP) of $y(\boldsymbol{x}_0)$ (Cressie (1993), Chapter 3 and Santner et al. (2003), Chapter 3). The conditional variance $\sigma_Z^2(1 - \boldsymbol{r}_0^t \boldsymbol{R}_0^{-1}\boldsymbol{r}_0)$ is a measure of the predictive uncertainty.

When components of $\boldsymbol{\phi}$ are unknown, one estimates them and plugs them into (1.6). The frequentist prediction is the mean of the distribution $\left[Y(\boldsymbol{x}_0)|\boldsymbol{y}^s, \widehat{\boldsymbol{\phi}}\right]$. On the other hand, the Bayesian predictive distribution is $[Y(\boldsymbol{x}_0)|\boldsymbol{y}^s]$, where one puts a prior, $[\boldsymbol{\phi}]$, on $\boldsymbol{\phi}$. One can numerically approximate $[Y(\boldsymbol{x}_0)|\boldsymbol{y}^s]$ by simulating $[\boldsymbol{\phi}|\boldsymbol{y}^s]$ and approximating the integral

$$[Y(\boldsymbol{x}_0)|\boldsymbol{y}^s] = \int [Y(\boldsymbol{x}_0)|\boldsymbol{y}^s, \boldsymbol{\phi}][\boldsymbol{\phi}|\boldsymbol{y}^s] d\boldsymbol{\phi} \tag{1.7}$$

by

$$\sum_{i=1}^{m} \left[Y(\boldsymbol{x}_0)|\boldsymbol{y}^s, \boldsymbol{\phi}^{(i)}\right] / m,$$

where m is the number of draws from $[\boldsymbol{\phi}|\boldsymbol{y}^s]$, and $\boldsymbol{\phi}^{(i)}$ denotes the ith draw from $[\boldsymbol{\phi}|\boldsymbol{y}^s]$. The Law of Large Number guarantees that the approximation converges to the true value of $[y(\boldsymbol{x}_0)|\boldsymbol{y}^s]$ with a sufficiently large m. In this thesis, we use the Bayesian predictive method. Without other specification, we regard the mean of $[Y(\boldsymbol{x}_0)|\boldsymbol{y}^s]$ as the predictor of $y(\boldsymbol{x}_0)$.

In applications, it is necessary to implement a numerical approach, together with a statistical model, to determine the posterior distribution $[\boldsymbol{\phi}|\boldsymbol{y}^s]$. We implement a Markov chain Monte Carlo sampling scheme, the Metropolis-Hastings (MH) algorithm in our examples.

1.5 The Metropolis-Hastings (M-H) Sampling Algorithm

The following procedure produces draws $\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(m)}, \phi^{(m+1)}$ that converge to the posterior distribution $[\boldsymbol{\phi}|\boldsymbol{y}^s]$ as $m \to \infty$. For a real-valued generic parameter ϕ,

Initialization Initialize ϕ to be $\phi^{(0)}$ and set $m = 1$.

Update Given $\phi^{(m)}$, the mth value in the sequence, generate a candidate $\phi^\star$ from a symmetric proposal distribution $f(\cdot \mid \phi^{(m)})$, i.e., $f(\phi^{(m)} \mid \phi^\star) = f(\phi^\star \mid \phi^{(m)})$. Set

$$\phi^{(m+1)} = \begin{cases} \phi^\star, & \text{with probability } \alpha \\ \phi^{(m)}, & \text{with probability } 1-\alpha \end{cases} \tag{1.8}$$

where

$$\alpha = min\left\{1, \frac{[\phi^\star \mid \boldsymbol{y}^s]}{[\phi^{(m)} \mid \boldsymbol{y}^s]}\right\}.$$

Recursion Increase m by 1 and repeat the **Update** Step.

It is known that draws from this algorithm will converge to the posterior distribution (Robert and Casella (1999)). Notice that, in this algorithm $[\phi|\boldsymbol{y}^s]$ needs only be known up to a proportional constant. By (1.4), $\frac{[\phi^\star|\boldsymbol{y}^s]}{[\phi^{(m)}|\boldsymbol{y}^s]}$ is computed as $\frac{[\phi^\star][\boldsymbol{y}^s|\phi^\star]}{[\phi^m][\boldsymbol{y}^s|\phi^{(m)}]}$. We specify our choices of the proposal distributions in Chapter 2, 3, and 4.

The Metropolis-Hastings algorithm has a number of advantages that make it worth using. First, it can handle multiple parameters. For example, if there are p parameters, i.e., $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)^\top$, then the update step in the algorithm is

Update For $i = 1, 2, \ldots, p$ in turn, given $\boldsymbol{\phi}_i^{(m)} = (\phi_1^{(m+1)}, \ldots, \phi_{i-1}^{(m+1)}, \phi_i^{(m)}, \phi_{i+1}^{(m)}, \ldots, \phi_p^{(m)})^\top$, generate a trial $\phi_i^\star$ from a symmetric proposal distribution $f(\cdot \mid \phi_i^{(m)})$,

i.e., $f(\phi_i^{(m)} \mid \phi_i^\star) = f(\phi_i^\star \mid \phi_i^{(m)})$. Set

$$\phi_i^{(m+1)} = \begin{cases} \phi_i^\star, & \text{with probability } \alpha \\ \phi_i^{(m)}, & \text{with probability } 1-\alpha \end{cases} \tag{1.9}$$

where

$$\alpha = min \left\{ 1, \frac{[\phi_i^\star \mid \boldsymbol{y}^s, \phi_1^{(m+1)}, \ldots, \phi_{i-1}^{(m+1)}, \phi_{i+1}^{(m)}, \ldots, \phi_p^{(m)}]}{[\phi_i^{(m)} \mid \boldsymbol{y}^s, \phi_1^{(m+1)}, \ldots, \phi_{i-1}^{(m+1)}, \phi_{i+1}^{(m)}, \ldots, \phi_p^{(m)}]} \right\}.$$

Second, the M-H algorithm works for multivariate output as long as the likelihood function of $[\boldsymbol{y}^s|\boldsymbol{\phi}]$ is available. Third, this M-H algorithm is time efficient in our examples. For the examples in Chapter 2, 3, and 4, the programs typically take 2 to 3 minutes to take $10,000$ draws.

CHAPTER 2

PREDICTION FOR COMPUTER EXPERIMENTS HAVING QUANTITATIVE AND QUALITATIVE INPUT VARIABLES

2.1 Introduction

The goal of this chapter is to develop a predictive model for the output from a computer code having both quantitative and qualitative inputs. While there are numerous well-developed statistical models for predicting the output from computer codes when all inputs are quantitative (Sacks et al. (1989b), Currin et al. (1991), Santner et al. (2003), Fang et al. (2005)), there have been relatively few attempts to propose models for cases where there are both quantitative and qualitative inputs. Kennedy and O'Hagan (2000) proposed an autoregressive model to describe outputs from computer codes of different complexities and running times but having the same set of quantitative inputs. In their case, different codes correspond to different levels of speed and fidelity. Qian, Wu and Wu (2008) proposed a Gaussian stochastic process model for mixed quantitative and qualitative input settings based on linear combinations of independent Gaussian processes. McMillan, Sacks, Welch and Gao (1999) proposed a proportionality model that can be used for predicting the output from a physical experiment. Conti and Hagan (2006) developed a Bayesian

methodology, based on a separable covariance model for their multivariate Gaussian stochastic processes. Their methodology can be applied to predicting scalar outputs when there are both quantitative and qualitative inputs. Qian and Wu (2008) proposed a Bayesian model to combine outputs from a faster (coarse) code and a slower (more accurate) code.

This chapter assumes that the outputs corresponding to different levels of a qualitative input are draws from Gaussian stochastic processes having "similar" correlation structures and magnitudes of variation. The proposed model describes the "similarities" in the model parameters by an appropriate prior distribution.

The outline of this chapter is as follows: Section 2.2 describes a hierarchical Bayesian model and the prediction for the output from a computer experiment having an arbitrary number of quantitative inputs and *one* qualitative input. Section 2.3 generalizes this model to handle multiple qualitative inputs. Section 2.4 illustrates the method with examples and compares the predictor of our model with three competing predictors. Section 2.5 implements the proposed model to a computer experiment having two quantitative inputs and two qualitative inputs. Section 2.6 summarizes this chapter.

2.2 Prediction for Computer Experiments Having Quantitative Input(s) and One Qualitative Input

Suppose the inputs to a computer experiment are t and $\boldsymbol{x}$ where $t \in \{1, 2, ..., T\}$ is a nominal-valued qualitative input and $\boldsymbol{x}$ is a $d \times 1$ vector denoting d quantitative inputs. We assume $\boldsymbol{x} \in [0, 1]^d$ or can be so scaled. We let $y(t, \boldsymbol{x})$ denote the real-valued output for the inputs t and $\boldsymbol{x}$. For n computer runs, we place the inputs in an $n \times (1 + d)$ matrix whose ith row is $(t_i, \boldsymbol{x}_i^\top)$, for all $i = 1, \ldots, n$, so that the left-most

column corresponds to the qualitative input. We let $\boldsymbol{y}_n = (y(t_1, \boldsymbol{x}_1), \ldots, y(t_n, \boldsymbol{x}_n))^\top$ denote the corresponding $n \times 1$ response vector and $y_0 \equiv y(t_0, \boldsymbol{x}_0)$ denote an unknown output to be predicted.

The *Hierarchical Quantitative-Qualitative Variable* (HQQV) model we propose as the basis for prediction is a hierarchical Bayesian model. We regard the model parameters at different levels as independently and identically distributed (*i.i.d.*) draws from the prior distribution to be specified.

The parameters in the HQQV model are denoted by $\boldsymbol{\beta} = (\beta_1, ..., \beta_T)^\top$, $\boldsymbol{\sigma} = (\sigma_1^2, \ldots, \sigma_T^2)^\top$, and $\boldsymbol{\rho} = (\boldsymbol{\rho}_1^\top, \ldots, \boldsymbol{\rho}_T^\top)^\top$ where $\boldsymbol{\rho}_t = (\rho_{t1}, \ldots, \rho_{td})^\top$ with $0 \leq \rho_{tj} \leq 1$ for all $t \in \{1, \ldots, T\}$ and $j \in \{1, \ldots, d\}$. The first stage of the model, given ($\boldsymbol{\beta}$, $\boldsymbol{\sigma}$, $\boldsymbol{\rho}$), regards the output at $(t, \boldsymbol{x})$ as a realization of the stochastic process

$$Y(t, \boldsymbol{x}) | (\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}) \sim \beta_t + Z_t(\boldsymbol{x}), \tag{2.1}$$

where $\beta_t \in \mathbb{R}^1$ and $Z_1(\cdot), \ldots, Z_T(\cdot)$ are *independent* stationary Gaussian processes with means zero and variances $\sigma_1^2, \ldots, \sigma_T^2$, respectively. For two $d \times 1$ quantitative inputs $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, $Z_t(\boldsymbol{x}_a)$ and $Z_t(\boldsymbol{x}_b)$ have covariance $\sigma_t^2 R(\boldsymbol{x}_a - \boldsymbol{x}_b | \boldsymbol{\rho}_t)$ where for $0 \leq \rho_{tj} \leq 1$ for all (t, j) and a $d \times 1$ difference vector $(h_1, \ldots, h_d)^\top$,

$$R((h_1, \ldots, h_d)^\top | \boldsymbol{\rho}_t) = \prod_{j=1}^{d} \rho_{tj}^{h_j^2}. \tag{2.2}$$

The correlation structure in (2.2) is the *Gaussian correlation*. The sample paths for this covariance structure are infinitely differentiable (Parzen (1967), Adler (1981)). Thus, this stage assumes that the outputs at each level t can be well-approximated by a smooth function of the quantitative inputs.

We construct higher stages of the Bayesian hierarchical model with the idea that the parameters should induce similarities of the responses at different levels of the

qualitative input. By setting the parameters at different levels to have the same prior distribution, we expect the predictions to borrow information from all the training data. We state our priors for $\boldsymbol{\beta}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\rho}$ next.

To construct meaningful priors, we standardize the outputs at each level of the qualitative input so that the outputs at each level have sample mean 0 and sample variance 1. The regression parameters $\beta_1, \ldots, \beta_T$ are taken to be *i.i.d.* with the standard non-informative prior distribution, which is proportional to 1. The variance parameters $\sigma_1^2, \ldots, \sigma_T^2$ are taken to be *i.i.d.* with informative Inverse Gamma(α, γ) where $\alpha > 0$ and $\gamma > 0$ are known so that $E(\sigma_t^2) = 1/[\gamma(\alpha - 1)]$ and $Var(\sigma_t^2) = 1/[\gamma^2(\alpha - 1)^2(\alpha - 2)]$ for $t \in \{1, \ldots, T\}$. Henceforth we denote the Inverse Gamma distribution by IG$(\cdot, \cdot)$. Specifically, the prior for $\sigma_1^2, \ldots, \sigma_T^2$, in the examples of this article, is taken to be IG$(5, 0.2)$, whose 95% symmetric probability interval is about $(0.49, 3.08)$. This prior has its mean and median close to 1 and allows the variance parameters to deviate some, but not greatly, from 1.

The prior distributions of $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_T$ are important in order for the predictions at any level of the qualitative input to be able to borrow information from the data at the other levels. With the assumption that the responses at all the levels have similar correlation structures, we construct an informative prior for each correlation parameter next. Recall from Chapter 1 that as ρ_{tj} increases to 1, the jth element of the quantitative input $\boldsymbol{x}$, x_j, has less impact on each of $y(t, \cdot)$ and $y(t, \cdot)$ is smoother in the jth dimension (Sacks et al. (1989b) and Santner et al. (2003), Chapter). We quantify the idea that the effects of x_j on $y(1, \cdot), \ldots, y(t, \cdot)$ are similar by assuming that $\rho_{1j}, \ldots, \rho_{Tj}$ are independently and identically Beta distributed with parameters $\alpha_j > 0$ and $\gamma_j > 0$ for all $j \in \{1, \ldots, d\}$. The beta distribution is denoted by

Be$(\cdot,\cdot)$ henceforth. The extent to which the effects of x_j on each of $y(1,\cdot),\ldots,y(t,\cdot)$ are similar depends on the mean and the variance of the Be(α_j,γ_j). This section proposes an empirical prior for the correlation parameters. This empirical prior is similar in spirit to the Uniform Shrinkage prior in Christiansen and Morris (1997) and Wallstrom (2007). Below we describe a two-step procedure for obtaining our empirical estimation of (α_j,γ_j).

1. Estimate ρ_{tj} using the training data in the level t for all $t\in\{1,\ldots,T\}$. Use the REstricted Maximum Likelihood (REML) estimation developed in Patterson and Thompson (1971). Let $\widehat{\rho}_{tj}$ denote the estimated ρ_{tj}.

2. Let the mean of Be(α_j,γ_j) be the *maximum* of $\widehat{\rho}_{1j},\ldots,\widehat{\rho}_{Tj}$ and have a lower bound 0.005 and an upper bound 0.995. Let the variance of Be(α_j,γ_j) be the sample variance of $\widehat{\rho}_{1j},\ldots,\widehat{\rho}_{Tj}$ but no larger than 0.004; i.e., we select α_j and γ_j to satisfy

$$\frac{\alpha_j}{\alpha_j+\gamma_j}=\text{median}\{0.005,M,0.995\} \tag{2.3}$$

and

$$\frac{\alpha_j\gamma_j}{(\alpha_j+\gamma_j)^2(\alpha_j+\gamma_j+1)}=\min\{s_j^2,0.004\}, \tag{2.4}$$

where $M=\max_{1\le t\le T}\{\widehat{\rho}_{tj}\}$ and $(T-1)s_j^2=\sum_{t=1}^{T}(\widehat{\rho}_{tj}-\overline{\widehat{\rho}_{\cdot j}})^2$ with $\overline{\widehat{\rho}_{\cdot j}}=\sum_{t=1}^{T}\widehat{\rho}_{tj}/T$.

The idea behind this shrinkage prior for $\rho_{1j},\ldots,\rho_{Tj}$ is as follows: When the design of the computer experiment is not space-filling or when the number of the training data points in level t is different from the numbers of points in the other levels, an estimate of ρ_{tj} can be close to zero. If $\widehat{\rho}_{tj}$ is near zero, the prediction of the output in level t will converge quickly, in the dimension j of the quantitative input, to the process

mean and thus the prediction errors can be undesirably large (see Santner et al. (2003) and Joseph (2006) for more details). With the assumption that the correlation structures of the processes at all the levels are similar, we avoid this problem by letting $M = \max_{1\le t\le T}\{\widehat{\rho}_{tj}\}$ be the mean of $\mathrm{Be}(\alpha_j, \gamma_j)$ when $M \in [0.005, 0.995]$. Furthermore, we let the variance of $\mathrm{Be}(\alpha_j, \gamma_j)$ be s_j^2 and have the upper bound 0.004 so that the estimates of the correlation parameters in the jth dimension are quantitatively similar (i.e., $\widehat{\rho}_{t_1 j} - \widehat{\rho}_{t_2 j} < 0.35$ for all $t_1, t_2 \in \{1, \ldots, T\}$). The other reason for setting the lower and upper bounds is that when the mean is in the interval $[0.005, 0.995]$ and the variance is less than 0.004, α_j and γ_j satisfy $\alpha_j > 0$ and $\gamma_j > 0$ so that a valid beta distribution is specified.

After specifying the priors, we further assume that $\boldsymbol{\beta}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\rho}$ are independent. The joint prior density of $\boldsymbol{\beta}, \boldsymbol{\sigma}$, and $\boldsymbol{\rho}$ can therefore be computed as the product of their prior densities. Based on the model and the prior distribution, we describe the prediction of unknown outputs next.

We attempt to predict $y_0 \equiv y(t_0, \boldsymbol{x}_0)$ given output $\boldsymbol{y}_n$, where $(t_0, \boldsymbol{x}_0)$ is assumed to be a new input with $t_0 \in \{1, \ldots, T\}$ and $\boldsymbol{x}_0 \in [0,1]^d$. We let the square bracket notation $[X|Y]$ denote the conditional distribution of X given Y and $[X]$ denote the (marginal) distribution of X. As in Section 1.4.3, the predictive distribution of $Y_0 \equiv Y(t_0, \boldsymbol{x}_0)$ is the conditional distribution $[Y_0|\boldsymbol{Y}_n]$ obtained from the prior distribution $[\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}]$ and the multivariate distribution

$$\left[\begin{pmatrix} Y_0 \\ \boldsymbol{Y}_n \end{pmatrix}\middle| \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}\right] \sim N\left(\begin{pmatrix} \boldsymbol{f}_0^\top \\ \boldsymbol{F} \end{pmatrix}\boldsymbol{\beta}, \begin{pmatrix} \Sigma_{y0} & \boldsymbol{\Sigma}_{0n}^\top \\ \boldsymbol{\Sigma}_{0n} & \boldsymbol{\Sigma}_{yn} \end{pmatrix}\right), \tag{2.5}$$

where $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n)^\top$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_T)^\top$. Here $\boldsymbol{f}_i$ is a 0/1 vector satisfying $\boldsymbol{f}_i^\top \boldsymbol{\beta} = \beta_{t_i}$ for all $i \in \{0, 1, \ldots, n\}$. Each covariance component in the Normal distribution in (2.5) denotes the covariance of the corresponding component of $(Y_0, \boldsymbol{Y}_n)$.

Thus $\Sigma_{y0} = \sigma_{t_0}^2$ is a scalar and Σ_{yn} is the $n \times n$ covariance matrix of $\boldsymbol{Y}_n$. The (i,j)th element of $\boldsymbol{\Sigma}_{yn}$ is equal to 0 if $t_i \neq t_j$. The (i,j)th element is equal to $\sigma_t^2 R(\boldsymbol{x}_i - \boldsymbol{x}_j | \boldsymbol{\rho}_{t_j})$ if $t_i = t_j$. Finally, $\boldsymbol{\Sigma}_{0n}$ is an $n \times 1$ vector with the jth element equal to 0 if $t_0 \neq t_j$ and to $\sigma_t^2 R(\boldsymbol{x}_0 - \boldsymbol{x}_j | \boldsymbol{\rho}_{t_j})$ if $t_0 = t_j$.

Given the parameters ($\boldsymbol{\beta}$, $\boldsymbol{\sigma}$, $\boldsymbol{\rho}$) and the training data $\boldsymbol{y}_n$ we can compute the conditional mean and the conditional variance of Y_0 given $\boldsymbol{y}_n$ as

$$E(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}) = \boldsymbol{f}_0^\top \boldsymbol{\beta} + \boldsymbol{\Sigma}_{0n}^\top \boldsymbol{\Sigma}_{yn}^{-1} (\boldsymbol{y}_n - \boldsymbol{F}\boldsymbol{\beta}) \tag{2.6}$$

and

$$Var(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}) = \Sigma_{y0} - \boldsymbol{\Sigma}_{0n}^\top \boldsymbol{\Sigma}_{yn}^{-1} \boldsymbol{\Sigma}_{0n}. \tag{2.7}$$

The Minimum Variance Unbiased predictor of $y(t_0, \boldsymbol{x}_0)$ is

$$\widehat{y}^{HQV}(t_0, \boldsymbol{x}_0) = E(Y_0 | \boldsymbol{y}_n) = E\left[E(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})\right] \tag{2.8}$$

(Santner et al. (2003), Chapter). As a measure of the predictive uncertainty, the variance of Y_0 given $\boldsymbol{y}_n$ is

$$Var(Y_0 | \boldsymbol{y}_n) = Var\left[E(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})\right] + E\left[Var(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})\right]. \tag{2.9}$$

To compute (2.8) and (2.9) numerically, we take draws from $[\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho} | \boldsymbol{y}_n]$. Then, using each draw of $(\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$, we estimate $E(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ and $Var(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ by applying (2.6) and (2.7). Thus, we obtain two samples. One contains estimates of $E(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ and the other contains estimates of $Var(Y_0 | \boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$. Using (2.8) and (2.9), the Strong Law of Large Numbers guarantees that the appropriate combinations of the sample means and the sample variances converge to $E(Y_0 | \boldsymbol{y}_n)$ and $Var\left(Y_0 | \boldsymbol{y}_n\right)$ almost everywhere.

To carry out this computation, we use the Metropolis-Hastings algorithm to draw values from $[\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}|\ \boldsymbol{y}_n]$ and then follow (2.8) and (2.9) to evaluate $E(Y_0|\boldsymbol{y}_n)$ and $Var(Y_0|\boldsymbol{y}_n)$. (See Higdon, Kennedy, Cavendish, Cafeo and Ryne (2004) and Higdon, Williams, Moore, McKay and Keller-McNulty (2005) for more detailed descriptions of the Metropolis-Hastings algorithm.) Our numerical setting of the algorithm is summarized below. For all $t \in \{1, \ldots, T\}$ and $j \in \{1, \ldots, d\}$,

- The initial value of β_t is taken to be 0. An updated β_t is a random draw from a uniform proposal distribution with the mean equal to the previous value of β_t and the range 0.5. As the algorithm iterates, β_t has no enforced upper bound nor lower bound.

- The initial value of σ_t^2 is taken to be 1. An updated σ_t^2 is a random draw from a uniform proposal distribution with the mean equal to the previous value of σ_t^2 and the range 0.1. As the algorithm iterates, σ_t^2 has the enforced lower bound 0 but no enforced upper bound. (If a draw of σ_t^2 is smaller than 0, the posterior density of that draw is 0.)

- The initial value of ρ_{tj} is taken to be the median of $\{0.005, \max_{1\le t\le T}\{\widehat{\rho}_{tj}\}, 0.995\}$. We parametrize ρ_{tj} by θ_{tj} where $\rho_{tj} = e^{-\theta_{tj}/4}$. An updated θ_{tj} is a random draw from a uniform proposal distribution with the mean equal to the previous value of θ_{tj} and the range 0.05. As the algorithm iterates, ρ_{tj} has the lower and upper bounds 0 and 1. (If a draw of ρ_{tj} is smaller than 0 or bigger than 1, the posterior density of that draw is 0.)

We have found that, in all the examples in Section 2.4, the acceptance rates of the parameters were between 0.5 and 0.9. For the data sets we have used, there has

been no convergence issues. Moreover, this approach is computationally efficient. Our MATLAB implementation requires only about 3 minutes on a 2.8GHz PC to compute 5000 burn-in and 10,000 production draws of $(\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\rho})$ and to predict 1580 outputs when there are two quantitative inputs, one qualitative input with four levels, and 20 training data points per level.

2.3 Prediction for Computer Experiments Having Quantitative Input(s) and Multiple Qualitative Inputs

A simple method to deal with multiple qualitative inputs is to regard the combinations of the values of the qualitative inputs as distinct values of a new single qualitative input variable. Philosophically, this is equivalent to including all the main and interaction effects of all the qualitative inputs. For example, if there are three qualitative inputs having 3, 3, and 2 levels, respectively, one can construct a single qualitative variable having $18 = 3 \times 3 \times 2$ levels where each level of the new qualitative input variable corresponds to a combination of the original three qualitative inputs. Generally, if there are K qualitative inputs and the kth input has T_k levels for all $k \in \{1, \ldots, K\}$, then the new qualitative input variable will have $\prod_{k=1}^{K} T_k$ levels and the HQQV model with a d-dimensional quantitative input and this new qualitative input has $(d+2) \times \prod_{k=1}^{K} T_k$ parameters (T regression parameters, T variance parameters, and $d \times T$ correlation parameters).

This method has at least two limitations. First, the number of parameters $(d+2) \times \prod_{k=1}^{K} T_k$ can be extremely large if K is big. Second, one may expect that as the number of identical components in $\boldsymbol{t}_a$ and $\boldsymbol{t}_b$ increases, the correlation between the two responses $y(\boldsymbol{t}_a, \boldsymbol{x}_a)$ and $y(\boldsymbol{t}_b, \boldsymbol{x}_b)$ will be increasing, where $\boldsymbol{t}_a = (t_a^{(1)}, \ldots, t_a^{(K)})^\top$ and $\boldsymbol{t}_b = (t_b^{(1)}, \ldots, t_b^{(K)})^\top$ denote two multiple qualitative inputs and $\boldsymbol{x}_a, \boldsymbol{x}_b \in [0,1]^d$

denote two quantitative inputs. The method above does not have this feature. To overcome these limitations, we propose a *multivariate HQQV model* next.

The idea of the multivariate HQQV model is to build a stochastic process for each qualitative input factor and to regard the response as coming from the sum of these processes. Specifically, suppose a computer code has K qualitative inputs (where $K \geq 1$) and the input has the form $(\boldsymbol{t}, \boldsymbol{x})$ where $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ denotes d quantitative inputs in $[0,1]^d$ and $\boldsymbol{t} = (t^{(1)}, \ldots, t^{(K)})^\top$ denotes the vector of K qualitative inputs where $t^{(k)} \in \{1, \ldots, T_k\}$ for all $k \in \{1, \ldots, K\}$. We let $y(\boldsymbol{t}, \boldsymbol{x})$ denote the corresponding real-valued output. To simplify the discussion, we first consider the case $K = 2$ and then generalize K to be any positive integer.

When $K = 2$, the multivariate HQQV model regards $y(\boldsymbol{t}, \boldsymbol{x}) = y(t^{(1)}, t^{(2)}, \boldsymbol{x})$ as a realization from a stochastic process $Y(t^{(1)}, t^{(2)}, \boldsymbol{x})$. The parameters corresponding to both $k = 1$ and $k = 2$ in this multivariate HQQV model have structures comparable to the structure of the parameters in the HQQV model in Section 2 except that a superscript (k) is used to denote the kth qualitative input. Corresponding to both $k = 1$ and 2, the parameters are $\boldsymbol{\beta}^{(k)} = (\beta_1^{(k)}, \ldots, \beta_{T_k}^{(k)})^\top$, $\boldsymbol{\sigma}^{(k)} = (\sigma_1^{(k)2}, \ldots, \sigma_{T_k}^{(k)2})^\top$, and $\boldsymbol{\rho}^{(k)} = ({\boldsymbol{\rho}_1^{(k)}}^\top, \ldots, {\boldsymbol{\rho}_{T_k}^{(k)}}^\top)^\top = (\rho_{11}^{(k)}, \ldots, \rho_{1d}^{(k)}, \ldots, \rho_{T_k 1}^{(k)}, \ldots, \rho_{T_k d}^{(k)})^\top$. Then given $(\boldsymbol{\beta}^{(1)}, \boldsymbol{\sigma}^{(1)}, \boldsymbol{\rho}^{(1)})$ and $(\boldsymbol{\beta}^{(2)}, \boldsymbol{\sigma}^{(2)}, \boldsymbol{\rho}^{(2)})$, we model $Y(t^{(1)}, t^{(2)}, \boldsymbol{x})$ as

$$\begin{aligned} Y(t^{(1)}, t^{(2)}, \boldsymbol{x}) | (\boldsymbol{\beta}^{(1)}, \boldsymbol{\sigma}^{(1)}, \boldsymbol{\rho}^{(1)}, \boldsymbol{\beta}^{(2)}, \boldsymbol{\sigma}^{(2)}, \boldsymbol{\rho}^{(2)}) \quad &\sim \\ Y_1(t^{(1)}, \boldsymbol{x}) | (\boldsymbol{\beta}^{(1)}, \boldsymbol{\sigma}^{(1)}, \boldsymbol{\rho}^{(1)}) \quad &+ \quad Y_2(t^{(2)}, \boldsymbol{x}) | (\boldsymbol{\beta}^{(2)}, \boldsymbol{\sigma}^{(2)}, \boldsymbol{\rho}^{(2)}), \end{aligned} \tag{2.10}$$

where

$$Y_k(t, \boldsymbol{x}) | (\boldsymbol{\beta}^{(k)}, \boldsymbol{\sigma}^{(k)}, \boldsymbol{\rho}^{(k)}) \sim \beta_t^{(k)} + Z_{k,t}(\boldsymbol{x}) \tag{2.11}$$

with $\beta_t^{(k)}$ being an unknown constant, $Z_{k,1}(\cdot), \ldots, Z_{k,T_k}(\cdot)$ being T_k *independent* Gaussian processes having zero means and variances ${\sigma_1^{(k)}}^2, \ldots, {\sigma_{T_k}^{(k)}}^2$, and $Z_{1,t^{(1)}}(\cdot)$ and $Z_{2,t^{(2)}}(\cdot)$ being also independent. For two quantitative inputs $\boldsymbol{x}_a \in [0,1]^d$ and $\boldsymbol{x}_b \in [0,1]^d$, the covariance between $Z_{k,t}(\boldsymbol{x}_a)$ and $Z_{k,t}(\boldsymbol{x}_b)$ is $\sigma_t^{(k)2} R(\boldsymbol{x}_a - \boldsymbol{x}_b | \boldsymbol{\rho}_t^{(k)})$, where $R(\cdot|\boldsymbol{\rho}_t^{(k)})$ is the *Gaussian correlation* in (2.2). As a result, given $(\boldsymbol{\beta}^{(1)}, \boldsymbol{\sigma}^{(1)}, \boldsymbol{\rho}^{(1)})$ and $(\boldsymbol{\beta}^{(2)}, \boldsymbol{\sigma}^{(2)}, \boldsymbol{\rho}^{(2)})$, $Y(t^{(1)}, t^{(2)}, \boldsymbol{x})$ is modeled as a Gaussian process with mean $\beta_{t^{(1)}}^{(1)} + \beta_{t^{(2)}}^{(2)}$, and for any two inputs $(\boldsymbol{t}^a, \boldsymbol{x}^a) = (t_a^{(1)}, t_a^{(2)}, \boldsymbol{x}_a)$ and $(\boldsymbol{t}^b, \boldsymbol{x}^b) = (t_b^{(1)}, t_b^{(2)}, \boldsymbol{x}_b)$, the covariance between $Y(t_a^{(1)}, t_a^{(2)}, \boldsymbol{x}_a)$ and $Y(t_b^{(1)}, t_b^{(2)}, \boldsymbol{x}_b)$ is

$$\begin{aligned} Cov(Y(t_a^{(1)}, t_a^{(2)}, \boldsymbol{x}_a), Y(t_b^{(1)}, t_b^{(2)}, \boldsymbol{x}_b)) &= \\ \sigma^2_{t_a^{(1)}} R(\boldsymbol{x}_a - \boldsymbol{x}_b | \boldsymbol{\rho}_{t_a^{(1)}}) I_0(t_a^{(1)} - t_b^{(1)}) &+ \sigma^2_{t_a^{(2)}} R(\boldsymbol{x}_a - \boldsymbol{x}_b | \boldsymbol{\rho}_{t_a^{(2)}}) I_0(t_a^{(2)} - t_b^{(2)}), \end{aligned} \tag{2.12}$$

where $I_0(\cdot)$ is a univariate indicator function such that $I_0(0) = 1$ and for all real valued $s \neq 0$, $I_0(s) = 0$.

When $K = 2$, we assume that all the model parameters have independent prior distributions. For both $k = 1$ and $k = 2$, the prior distributions of $(\boldsymbol{\beta}^{(k)}, \boldsymbol{\sigma}^{(k)}, \boldsymbol{\rho}^{(k)})$ are constructed in the same way as the priors of $(\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ in Section 2. Specifically, we standardize the outputs at each $(t^{(1)}, t^{(2)})$ combination to have mean 0 and variance 1. We take the regression parameters $\{\beta_t^{(k)} | t = 1, \ldots, T_k\}$ to be *i.i.d.* with standard non-informative prior proportional to 1. We take the variance parameters $\{\sigma_t^{(k)2} | t = 1, \ldots, T_k\}$ to be *i.i.d.* with the prior $\sigma_t^{(k)2} \sim \text{IG}(5, 0.2 \times 2)$, so that for all $t^{(1)} \in \{1, \ldots, T_1\}$ and $t^{(2)} \in \{1, \ldots, T_2\}$, $\sigma^{(1)2}_{t^{(1)}} + \sigma^{(2)2}_{t^{(2)}}$ is roughly 1 and the posterior draws of $\sigma^{(2)2}_{t^{(1)}} + \sigma^{(2)2}_{t^{(2)}}$ can deviate some, but not greatly, from 1.

Same as the prior for $\boldsymbol{\rho}$ in Section 2, we use an empirical prior, which is similar in spirit to the Uniform Shrinkage prior in Christiansen and Morris (1997) and Wallstrom (2007), for the correlation parameters $\boldsymbol{\rho}^{(1)}$ and $\boldsymbol{\rho}^{(2)}$. For all $j \in \{1, \ldots, d\}$, let

$\widehat{\rho}_{1j}^{(k)}, \ldots, \widehat{\rho}_{T_k j}^{(k)}$ denote the REML estimates of $\rho_{1j}^{(k)}, \ldots, \rho_{T_k j}^{(k)}$. The correlation parameters $\{\rho_{tj}^{(k)} | t \in \{1, \ldots, T_k\}\}$ are taken to be *i.i.d.* with $\text{Be}(\alpha_j^{(k)}, \gamma_j^{(k)})$. Similar to (2.3) and (2.4), the hyper-parameters $\alpha_j^{(k)}$ and $\gamma_j^{(k)}$ are computed using

$$\frac{\alpha_j^{(k)}}{\alpha_j^{(k)} + \gamma_j^{(k)}} = \text{median}\{0.005, M^{(k)}, 0.995\}$$

and

$$\frac{\alpha_j^{(k)} \gamma_j^{(k)}}{(\alpha_j^{(k)} + \gamma_j^{(k)})^2 (\alpha_j^{(k)} + \gamma_j^{(k)} + 1)} = \min\{s_j^{(k)2}, 0.004\},$$

where $M^{(k)} = \max_{1 \le t \le T_k} \{\widehat{\rho}_{tj}^{(k)}\}$ and $(T_k - 1) s_j^{(k)2} = \sum_{t=1}^{T_k} (\widehat{\rho}_{tj}^{(k)} - \overline{\widehat{\rho}_{\cdot j}^{(k)}})^2$ with $\overline{\widehat{\rho}_{\cdot j}^{(k)}} = \sum_{t=1}^{T_k} \widehat{\rho}_{tj} / T_k$.

Next, we describe how to generalize the above model and priors to computer experiments having d quantitative inputs and K qualitative inputs, where d and K can be any positive integers. Using the same additive structure as in (2.10), the model regards an observation $y(\boldsymbol{t}, \boldsymbol{x})$ as a realization of a random function $Y(\boldsymbol{t}, \boldsymbol{x})$. Given the model parameters $\boldsymbol{\beta} = ({\boldsymbol{\beta}^{(1)}}^\top, \ldots, {\boldsymbol{\beta}^{(K)}}^\top)^\top$, $\boldsymbol{\sigma} = ({\boldsymbol{\sigma}^{(1)}}^\top, \ldots, {\boldsymbol{\sigma}^{(K)}}^\top)^\top$, and $\boldsymbol{\rho} = ({\boldsymbol{\rho}^{(1)}}^\top, \ldots, {\boldsymbol{\rho}^{(K)}}^\top)^\top$ we model $Y(\boldsymbol{t}, \boldsymbol{x})$ as

$$Y(\boldsymbol{t}, \boldsymbol{x}) | (\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}) \sim \sum_{k=1}^{K} Y_k(t^{(k)}, \boldsymbol{x}) | (\boldsymbol{\beta}^{(k)}, \boldsymbol{\sigma}^{(k)}, \boldsymbol{\rho}^{(k)}), \tag{2.13}$$

where $Y_1(\cdot, \cdot), \ldots, Y_K(\cdot, \cdot)$ are K *independent* Gaussian processes. For all $k = 1, \ldots, K$, given the parameters corresponding to the kth qualitative input $\boldsymbol{\beta}^{(k)} = (\beta_1^{(k)}, \ldots, \beta_{T_k}^{(k)})^\top$, $\boldsymbol{\sigma}^{(k)} = ({\sigma_1^{(k)}}^2, \ldots, \sigma_{T_k}^{(k)2})^\top$, and $\boldsymbol{\rho}^{(k)} = ({\boldsymbol{\rho}_1^{(k)}}^\top, \ldots, {\boldsymbol{\rho}_{T_k}^{(k)}}^\top)^\top = (\rho_{11}^{(k)}, \ldots, \rho_{1d}^{(k)}, \ldots, \rho_{T_k 1}^{(k)}, \ldots, \rho_{T_k d}^{(k)})^\top$, $Y_k(t^{(k)}, \boldsymbol{x})$ is modeled as (2.11). Thus, $Y(\boldsymbol{t}, \boldsymbol{x})$ is a Gaussian stochastic process with mean $\sum_{k=1}^{K} \beta_{t^{(k)}}^{(k)}$ and covariance between $Y(\boldsymbol{t}_a, \boldsymbol{x}_a)$ and $Y(\boldsymbol{t}_b, \boldsymbol{x}_b)$

$$Cov(Y(\boldsymbol{t}_a, \boldsymbol{x}_a), Y(\boldsymbol{t}_b, \boldsymbol{x}_b)) = \sum_{k=1}^{K} \sigma_{t_a^{(k)}}^2 R(\boldsymbol{x}_a - \boldsymbol{x}_b | \boldsymbol{\rho}_{t_a^{(k)}}) I_0(t_a^{(k)} - t_b^{(k)}), \tag{2.14}$$

where $\boldsymbol{t}_a = (t_a^{(1)}, \ldots, t_a^{(K)})$ and $\boldsymbol{t}_b = (t_b^{(1)}, \ldots, t_b^{(K)})$ with $t_a^{(k)}, t_b^{(k)} \in \{1, \ldots, T_k\}$ and $\boldsymbol{x}_a, \boldsymbol{x}_b \in [0,1]^d$. All the model parameters are set to have *independent* prior distributions and for all $k = 1, \ldots, K$, the prior distributions of $(\boldsymbol{\beta}^{(k)}, \boldsymbol{\sigma}^{(k)}, \boldsymbol{\rho}^{(k)})$ are constructed in the same way as the priors of $(\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ in Section 2. Specifically, we standardize the responses at each value of $\boldsymbol{t}$ to have mean 0 and variance 1. Then we set the priors of $(\boldsymbol{\beta}^{(k)}, \boldsymbol{\sigma}^{(k)}, \boldsymbol{\rho}^{(k)})$ (where K is any positive integer and $k = 1, \ldots, K$) to be the same as the above priors of $(\boldsymbol{\beta}^{(k)}, \boldsymbol{\sigma}^{(k)}, \boldsymbol{\rho}^{(k)})$ (where $K = 2$ and $k = 1, 2$) except that we let the prior of $\{\sigma_t^{(k)2} | t = 1, \ldots, T_k\}$ be i.i.d. IG$(5, 0.2 \times K)$ so that for all $k = 1, \ldots, K$ and $t^{(k)} \in \{1, \ldots, T_k\}$, $\sum_{k=1}^K \sigma_{t^{(k)}}^{(k)2}$ is roughly 1 and the posterior draws of $\sum_{k=1}^K \sigma_{t^{(k)}}^{(k)2}$ can deviate some, but not greatly, from 1. It is worth noting that compared with the simple method that uses a qualitative input having $\prod_{k=1}^K T_k$ levels to replace the K qualitative input variables and requires $(d+2)\prod_{k=1}^K T_k$ parameters, the multiple HQQV model reduces the number of parameters from $(d+2)\prod_{k=1}^K T_k$ to $(d+2)\sum_{k=1}^K T_k$.

To predict an unknown response $y(\boldsymbol{t}_0, \boldsymbol{x}_0)$ given the response vector $\boldsymbol{y}_n = (y(\boldsymbol{t}_1, \boldsymbol{x}_1), \ldots, y(\boldsymbol{t}_n, \boldsymbol{x}_n))^\top$, we first simulate the posterior distribution of the model parameters using the Metropolis-Hastings algorithm and then use the draws from the simulation to approximate $\widehat{y}^{HQV}(\boldsymbol{t}_0, \boldsymbol{x}_0) = E(Y(\boldsymbol{t}_0, \boldsymbol{x}_0) | \boldsymbol{y}_n)$. Specifically, for all $k \in \{1, \ldots, K\}$, $t \in \{1, \ldots, T_k\}$ and $j \in \{1, \ldots, d\}$, the initial values of $\beta_t^{(k)}$, $\sigma_t^{(k)2}$, and $\rho_{tj}^{(k)}$ are taken to be 0, $1/K$, and the median of $\{0.005, \max_{1 \le t \le T_k}\{\widehat{\rho}_{tj}^{(k)}\}, 0.995\}$. We parametrize $\rho_{tj}^{(k)}$ by $\theta_{tj}^{(k)}$ where $\rho_{tj}^{(k)} = e^{-\theta_{tj}^{(k)}/4}$. Updated $\beta_t^{(k)}$, $\sigma_t^{(k)2}$, and $\theta_{tj}^{(k)}$ are drawn from uniform proposal distributions with the means being the previous values of $\beta_t^{(k)}$, $\sigma_t^{(k)2}$, and $\theta_{tj}^{(k)}$ and the ranges being 0.5, $0.1/K$, and 0.05, respectively. As the algorithm iterates, $\beta_t^{(k)}$

has no enforced upper bound nor lower bound, $\sigma_t^{(k)2}$ has the lower bound 0 but no enforced upper bound, and $\rho_{tj}^{(k)}$ has the lower bound 0 and the upper bound 1. Similar to (2.8) and (2.9), the predictor of $y(\boldsymbol{t}_0, \boldsymbol{x}_0)$ and a measure of the predictive uncertainty are $E(Y(\boldsymbol{t}_0, \boldsymbol{x}_0)|\boldsymbol{y}_n) = E\left[E(Y(\boldsymbol{t}_0, \boldsymbol{x}_0)|\boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})\right]$ and $Var(Y(\boldsymbol{t}_0, \boldsymbol{x}_0)|\boldsymbol{y}_n)$ $= Var\left[E(Y(\boldsymbol{t}_0, \boldsymbol{x}_0)|\boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})\right] + E\left[Var(Y(\boldsymbol{t}_0, \boldsymbol{x}_0)|\boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})\right]$, which can be approximated by plugging the draws of $(\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ into $E(Y(\boldsymbol{t}_0, \boldsymbol{x}_0)|\boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ and $Var(Y(\boldsymbol{t}_0, \boldsymbol{x}_0)|\boldsymbol{y}_n, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ and then applying the Law of Large Numbers.

2.4 Comparing the HQQV Predictor with Three Competing Predictors

In this section, we will compare the predictive accuracy of the HQQV predictor with three competing predictors in examples having one qualitative input. Before introducing examples, we first specify the competing predictors.

2.4.1 Competing Predictors

A Surface-wise Hierarchical Bayes Predictor

The first competing predictor we consider is a hierarchical Bayes predictor computed separately for each surface corresponding to a level of the qualitative input. This model is comparable with the HQQV model having $T = 1$. Specifically, we regard the output of a computer code as coming from a realization of the stochastic process

$$Y(t, \boldsymbol{x})|(\beta_t, \sigma_t^2, \boldsymbol{\rho}_t) \sim \beta_t + Z_t(\boldsymbol{x}), \tag{2.15}$$

where $Z_t(\boldsymbol{x})$ is a Gaussian stochastic process with mean zero, covariance σ_t^2, and correlation structure (2.2) with parameters $\boldsymbol{\rho}_t = (\rho_{t1}, \ldots, \rho_{td})'$. The stationary processes $Z_1(\cdot), \ldots, Z_T(\cdot)$ are independent. However, the higher stages of this model do not tie

together the processes modeling the outputs at different levels of the qualitative input. The second stage specifies that the means $\beta_1, \ldots, \beta_T$ are *i.i.d.* with non-informative prior proportional to 1, the variance parameters $\sigma_1^2, \ldots, \sigma_T^2$ are *i.i.d.* IG$(5, 0.2)$, and the correlation parameters $\{\rho_{tj}; t \in \{1, \ldots, T\}, j \in \{1, \ldots, d\}\}$ are independent with $\rho_{tj} \sim \text{Be}(\alpha_{tj}, \gamma_{tj})$. We determine the beta parameters by setting the mean and variance of $\text{Be}(\alpha_{tj}, \gamma_{tj})$ equal to the median of $\{0.005, \widehat{\rho}_{tj}, 0.995\}$ and 0.004, respectively, where $\widehat{\rho}_{tj}$ is the REML estimate (of ρ_{tj}) obtained using the training data at level t. We let $\widehat{y}^{SHB}(t_0, \boldsymbol{x}_0)$ denote the predictor of $y(t_0, \boldsymbol{x}_0)$, which is defined as the conditional mean of $Y(t_0, \boldsymbol{x}_0)$ given the training data at the level t_0, $\boldsymbol{y}_{n_{t_0}}$; i.e., $\widehat{y}^{SHB}(t_0, \boldsymbol{x}_0) = E(Y(t_0, \boldsymbol{x}_0)|\boldsymbol{y}_{n_{t_0}})$. We refer to this predictor as the "Surface-wise Hierarchical Bayes" (SHB) predictor.

An Autoregressive Predictor

Kennedy and O'Hagan (2000) described an autoregressive multivariate model applied to the output from several codes of increasing accuracies but for the same physical phenomenon. In this setup, the slowest computer code is the gold-standard whose outputs we wish to predict.

Below, we consider a $T = 3$ level application of the autoregressive model with a scalar quantitative input $x \in [0, 1]$ and so briefly describe the model in this setting. Let $y(1, x)$, $y(2, x)$, and $y(3, x)$ denote the output of the three codes where $y(3, x)$ is the gold-standard output, $y(2, x)$ is a less accurate version of $y(3, x)$, and $y(1, x)$ is less accurate than $y(2, x)$. Based on the training data from the three codes, the object is to predict $y(3, x_0)$, where $(3, x_0)$ denotes an untried input. The responses are viewed as coming from a three-variate process $(Y(1, x), Y(2, x), Y(3, x))'$,

where $Y(1,x)$, $Y(2,x)$, and $Y(3,x)$ are modeled as linear combinations of three independent processes $Z_1(x)$, $Z_2(x)$, and $Z_3(x)$ to mimic a hierarchical autoregressive process; i.e., $Y(1,x)= Z_1(x)$, $Y(2,x) = \tau_1 Y(1,x) + Z_2(x) = \tau_1 Z_1(x) + Z_2(x)$, and $Y(3,x) = \tau_2 Y(2,x) + Z_3(x) = \tau_2\tau_1 Z_1(x) + \tau_2\ Z_2(x) + Z_3(x)$. Here, $Z_1(x)$, $Z_2(x)$, and $Z_3(x)$ are assumed to be *independent stationary* Gaussian Stochastic Processes having unknown constant means β_1, β_2, and β_3. For all $i, j \in \{1,2,3\}$ and $x_1, x_2 \in [0,1]$, the covariance between $Z_i(x_1)$ and $Z_i(x_2)$ is set to $\sigma_i^2 \times \rho_i^{(x_1-x_2)^2}$. Following Kennedy and O'Hagan (2000), we estimate $(\boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ by maximum likelihood estimate (MLE). Given the estimates $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\rho}})$ of $(\boldsymbol{\beta}, \boldsymbol{\tau}, \boldsymbol{\sigma}, \boldsymbol{\rho})$ and the training data $\boldsymbol{y}_n$, the predictor of $y(3,x_0)$ is the mean of the conditional normal distribution $[Y(3,x_0)|\boldsymbol{y}_n, \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\rho}}]$, i.e., $\widehat{y}^{KOH}(3,x_0) = E(Y(3,x_0)|\boldsymbol{y}_n, \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\sigma}}, \widehat{\boldsymbol{\rho}})$.

Note that the autoregressive model treats the qualitative input as ordinal. While its intent is to perform prediction only for the final level of the qualitative input, the method can be used sequentially to predict the outputs at any level of the ordinal variable based on the "preceding" data. In the examples given in Section 4.3, we took the code at level $t = 3$ of the qualitative variable as the gold-standard and used *all* the data to predict the output at this level.

A Proportionality-based Predictor

Qian et al. (2008) regarded the training data as a draw from $[\boldsymbol{Y}_n|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\rho}, K(\cdot)]$ $\sim N(\boldsymbol{F}\boldsymbol{\beta}, \sigma^2\boldsymbol{R})$, where $\boldsymbol{F}$ denoted an $n \times q$ matrix and was a *known function* of the n inputs $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$, $\boldsymbol{\beta}$ denoted a $q \times 1$ vector of *unknown coefficients*, σ^2 denoted an unknown variance parameter, and $\boldsymbol{R}$ denoted an $n \times n$ correlation matrix with unknown correlation parameters. For two inputs $(t_1, \boldsymbol{x}_1)$ and $(t_2, \boldsymbol{x}_2)$, where $t_1, t_2 \in \{1, \ldots, T\}$ and $\boldsymbol{x}_1, \boldsymbol{x}_2 \in [0,1]^d$, the correlation between $Y(t_1, \boldsymbol{x}_1)$ and

$Y(t_2, \boldsymbol{x}_2)$ was denoted by $\prod_{i=1}^{d} \rho_i^{(x_{1i}-x_{2i})^2}$ if $t_1 = t_2$ and by $K(t_1, t_2) \times \prod_{i=1}^{d} \rho_i^{(x_{1i}-x_{2i})^2}$ for $0 \leq K(t_1, t_2) \leq 1$ if $t_1 \neq t_2$.

The proportionality-based predictor (PBP) regards the function $K(t_1, t_2)$ as an unknown constant (i.e., $K(t_1, t_2) \equiv \kappa$) and let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_T)'$, where $\beta_1, \ldots, \beta_T$ are the unknown constant means of the stochastic processes at levels $1, \ldots, T$. We let $\widehat{y}^{PBP}(t_0, \boldsymbol{x}_0) = E(Y(t_0, \boldsymbol{x}_0) | \boldsymbol{y}_n, \widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\rho}}, \widehat{\kappa})$ (the Empirical Best Linear Unbiased Predictor (EBLUP) of $y(t_0, \boldsymbol{x}_0)$) denote the proportionality-based predictor of $y(t_0, \boldsymbol{x}_0)$, where $\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\rho}}$, and $\widehat{\kappa}$ are the maximum likelihood estimators of $\boldsymbol{\beta}$, $\boldsymbol{\rho}$, and κ.

2.4.2 Comparison of $\widehat{y}^{SHB}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$

In this first example, we investigate the effect on the predictive accuracy of tying the processes together at different levels of the qualitative input in the idealized situation where the data are consistent with both the HQQV model assumption and the SHB model assumption. We then compare their predictive accuracies.

We generate the testing data sets following the steps described next. Each testing data set consists of $T = 4$ surfaces; each surface has 1600 data points. We let $\boldsymbol{x} \in [0, 1]^2$ denote the $d = 2$ continuous inputs.

Step 1 Set parameters $\beta_1 = \cdots = \beta_4 = 0$, $\sigma_1^2 = \ldots = \sigma_4^2 = 1$, and $\rho_1 = \ldots = \rho_4 = \rho$. The value of ρ is taken to be either 0.5 or 0.9 in our simulations.

Step 2 Generate an input set having 40×40 points by crossing $\{\frac{1}{80}, \frac{3}{80}, \ldots, \frac{79}{80}\}$ with the same 40 values. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{1600}$ denote the 1600 inputs. For each $t \in \{1, \ldots, 4\}$, simulate a vector $\boldsymbol{y}_{(t,1600)} = (y(t, \boldsymbol{x}_1), \ldots, y(t, \boldsymbol{x}_{1600}))'$ as a random sample of a 1600×1 multinormally distributed random vector $\boldsymbol{Y}_{(t,1600)} =$

$(Y(t, \boldsymbol{x}_1), \ldots, Y(t, \boldsymbol{x}_{1600}))'$ where $Y(t, \boldsymbol{x}_1), \ldots, Y(t, \boldsymbol{x}_{1600})$ have means 0 and variances 1 and $\boldsymbol{Y}_{(t,1600)}$ has the 1600×1600 correlation matrix with the Gaussian correlation $\rho^{(x_{i1}-x_{j1})^2}\rho^{(x_{i2}-x_{j2})^2}$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in [0,1]^2$. With the 1600 quantitative inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{1600}$, regard $y(t, \boldsymbol{x}_1)$, ..., $y(t, \boldsymbol{x}_{1600})$ as 1600 points on the tth simulated response surface.

Thus, these data satisfy the HQQV and SHB model assumptions. One can adjust the curvatures of the simulated surfaces by modifying ρ. The two panels in Figure 2.1 show two simulated surfaces. The surface on the left panel is with $\rho = 0.5$; the one on the right panel is with $\rho = 0.9$. We see that the simulated surface with a bigger ρ is smoother.

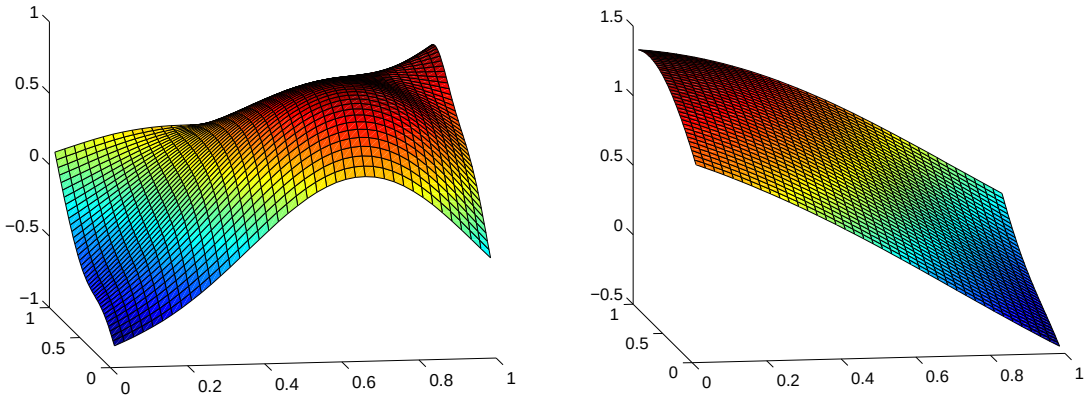

Figure 2.1: A simulated surface with $\rho = 0.5$ (the left panel) and a simulated surface with $\rho = 0.9$ (the right panel).

We generate the training data set by conducting a design for the quantitative inputs and acquiring the outputs at each level t from $\boldsymbol{y}_{(t,1600)}$. We construct a Maximin Latin Hypercube Design (Maximin LHD) having size n (McKay et al. (1979), Johnson

et al. (1990)). We let $n = 8$ or $n = 20$ and let the design inputs coincide with n points in the 40×40 input data set.

With a training data set having $4 \times n$ points, we predict the remaining $(1600 - n)$ outputs when $t = 1$. We measure the predictive accuracy using the *Root Mean Squared Prediction Error* (RMSPE) over the $(1600 - n)$ points. The RMSPE of a predictor $\widehat{y}(\cdot)$ is defined as

$$\sqrt{\frac{1}{1600 - n} \sum_{i=1}^{1600-n} (y(1, \boldsymbol{x}_i) - \widehat{y}(1, \boldsymbol{x}_i))^2}.$$

We considered four choices of (ρ, n) : $(0.5, 8)$, $(0.5, 20)$, $(0.9, 8)$, and $(0.9, 20)$. For each choice, we generated 100 data sets. For each data set, we predicted the $(1600-n)$ values of $y(1, \boldsymbol{x})$ using $\widehat{y}^{SHB}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ and then computed the RMSPEs of the two predictors. We thus obtained 100 pairs of the RMSPEs for each choice of (ρ, n). Each of the four panels in Figure 2.2 is a plot of the 100 pairs of the RMSPEs of $\widehat{y}^{SHB}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ for one of the four combinations. The average of the 100 RMSPEs of $\widehat{y}^{SHB}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ are 0.261 and 0.222 for $(\rho, n) = (0.5, 8)$, 0.066 and 0.072 for $(\rho, n) = (0.5, 20)$, 0.035 and 0.026 for $(\rho, n) = (0.9, 8)$, and 0.004 and 0.004 for $(\rho, n) = (0.9, 20)$.

Figure 2.2 and the average RMSPEs show that for both $\rho = 0.5$ and $\rho = 0.9$, the two predictors have comparable prediction errors when $n = 20$, but $\widehat{y}^{HQV}(\cdot)$ generally has significantly smaller prediction errors than $\widehat{y}^{SHB}(\cdot)$ when $n = 8$. It is worth noting that we have used values of ρ other than 0.5 and 0.9 to run the same procedure. We have found that, except for ρ near 0, e.g., $\rho \leq 0.05$ (where the observations are approximately coming from a noise process and thus accurate prediction of unknown outputs is not possible), the above mentioned contrast between the RMSPEs

of $\widehat{y}^{HQV}(\cdot)$ and $\widehat{y}^{SHB}(\cdot)$ holds. So $\widehat{y}^{HQV}(\cdot)$ has the advantage of borrowing information from the data at all the levels. Furthermore, this advantage is more obvious when the number of training data at the level for prediction is small.

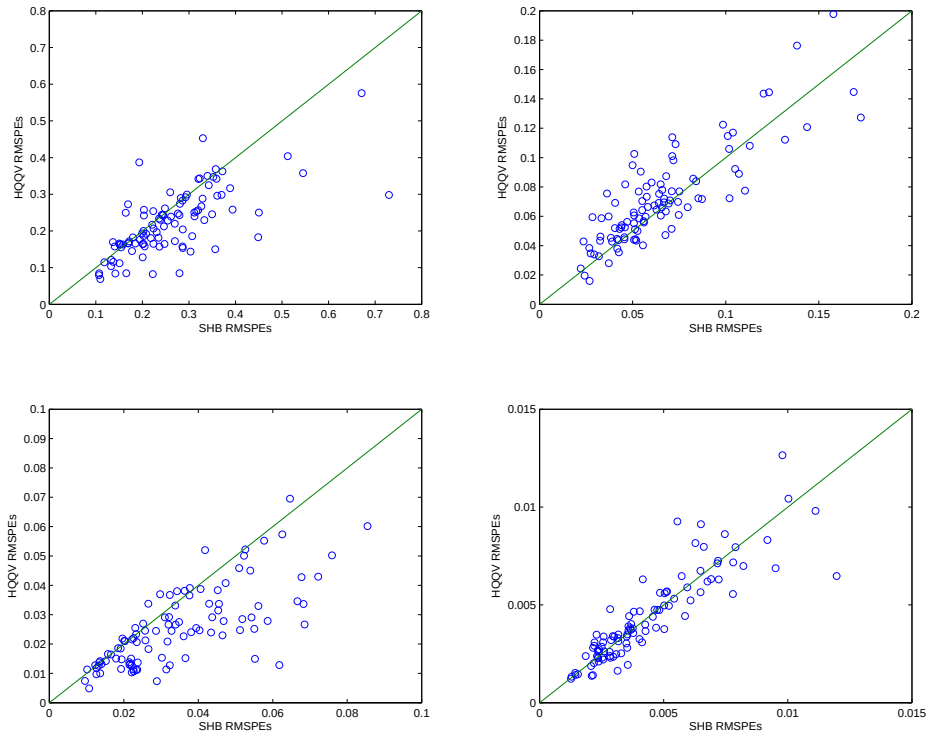

Figure 2.2: Four plots of the RMSPE comparisons of $\widehat{y}^{SHB}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ for 100 test surfaces. The upper left panel uses $(\rho, n) = (0.5, 8)$, the upper right panel uses $(\rho, n) = (0.5, 20)$, the lower left panel uses $(\rho, n) = (0.9, 8)$, and the lower right panel uses $(\rho, n) = (0.9, 20)$. In each panel, the horizontal axis corresponds to the RMSPE of $\widehat{y}^{SHB}(\cdot)$; the vertical axis corresponds to the RMSPE of $\widehat{y}^{HQV}(\cdot)$; the solid line is the 45 degree line passing through the origin; the circles denote the RMSPEs of $\widehat{y}^{HQV}(\cdot)$ against $\widehat{y}^{SHB}(\cdot)$; a circle below the 45 degree line indicates that $\widehat{y}^{HQV}(\cdot)$ has a smaller RMSPE than $\widehat{y}^{SHB}(\cdot)$.

2.4.3 Interpolation and Extrapolation Accuracies of $\widehat{y}^{SHB}(\cdot)$, $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{PBP}(\cdot)$, and $\widehat{y}^{HQV}(\cdot)$

In this second example, we compare $\widehat{y}^{SHB}(\cdot)$, $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{PBP}(\cdot)$, and $\widehat{y}^{HQV}(\cdot)$ in three cases, where the true data come from three quadratic curves; thus in each case, $T = 3$ and $d = 1$. We denote the three curves by $y(1, x) = b_{01} + b_{11}x + b_{21}x^2$, $y(2, x) = b_{02} + b_{12}x + b_{22}x^2$, and $y(3, x) = b_{03} + b_{13}x + b_{23}x^2$ with $x \in [0, 1]$. We observe $y(1, \cdot)$ and $y(2, \cdot)$ at five equally spaced inputs $\{0, 0.25, 0.5, 0.75, 1\}$. The third curve is observed only at three inputs $\{0.5, 0.75, 1\}$. We intentionally selected inputs at the boundary, e.g., $x = 0$ or 1, because in many situations, the responses at the boundary are of particular interest. For example, in Rawlinson et al. (2006) the internal and external rotational force of a knee prosthesis at the starting and ending positions of a gait cycle are of interest. We chose the quantitative input at $t = 3$ to be between 0.5 and 1 so that we can investigate both the interpolation and the extrapolation accuracies.

To compare the predictive accuracies, we first describe three processes (corresponding to the three cases) producing the true quadratic curves and the training data on the curves. For each process, the coefficients of the quadratic curves are drawn from independent normal distributions whose standard deviations are 0.01. The expected values of $(b_{01}, b_{02}, b_{03}, b_{11}, b_{12}, b_{13}, b_{21}, b_{22}, b_{23})$ are set to be $(1, 0, -1, 6, 4, 5, -6, -6, -6)$ (for Process 1), $(1, 0, -1, 0, 6, 5, 2, -6, -6)$ (for Process 2), and $(1, 0, -1, 6, 6, 6, -6, -6, -6)$ (for Process 3). Geometrically, the three curves drawn using Process 1 are all concave and differ only slightly in terms of their maxima and curvatures. One of the three curves drawn using Process 2 has substantially different trend, shape, curvature, and concavity than the other two curves. The three curves drawn using Process 3 are

nearly identical except for their intercepts. For each of the three processes, a draw of the three curves and the training data are depicted in Figure 2.3.

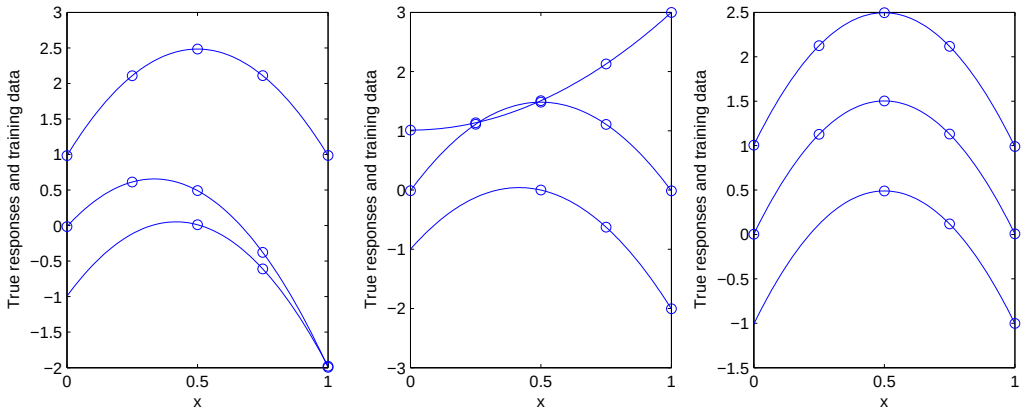

Figure 2.3: Plots of the true responses (solid curves) and the training data (circles) for one draw using Process 1 (the left panel), Process 2 (the middle panel), and Process 3 (the right panel).

With a known function $y(3, \cdot)$ generated by a process, the RMSPE of a predictor $\widehat{y}(\cdot)$ over n_{pred} points $x_1, x_2, \ldots, x_{n_{\text{pred}}}$ is

$$RMSPE = \sqrt{\frac{1}{n_{\text{pred}}} \sum_{i=1}^{n_{\text{pred}}} (\widehat{y}(3, x_i) - y(3, x_i))^2}. \tag{2.16}$$

To explore the *interpolation* accuracies, we let the inputs for prediction

$$\{x_1, x_2, \ldots, x_{n_{\text{pred}}}\} = \{0.5, 0.51, \ldots, 1.00\};$$

to explore the *extrapolation* accuracies, we let

$$\{x_1, x_2, \ldots, x_{n_{\text{pred}}}\} = \{0, 0.01, \ldots, 0.50\}.$$

We generated 30 true data sets for each process and computed the RMSPEs in (2.16). The boxplots in Figures 2.4 and 2.5 display the interpolation and extrapolation RMSPEs of the four predictors for each of the three processes.

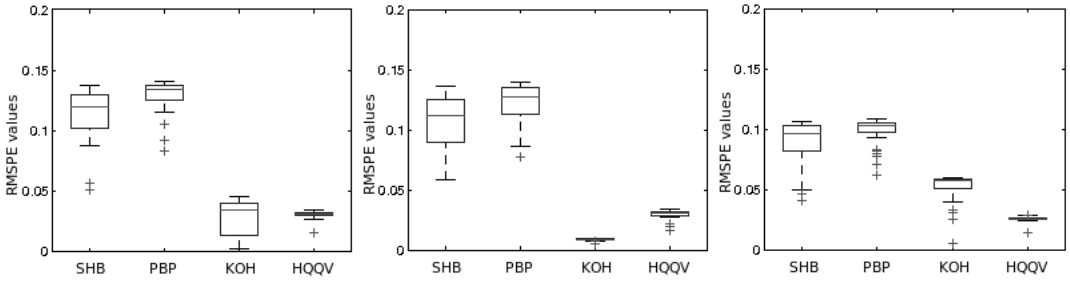

Figure 2.4: Boxplots of the interpolation RMSPEs of $\widehat{y}^{SHB}(\cdot)$, $\widehat{y}^{PBP}(\cdot)$, $\widehat{y}^{KOH}(\cdot)$, and $\widehat{y}^{HQV}(\cdot)$. The three panels correspond to Process 1 (the left panel), Process 2 (the middle panel), and Process 3 (the right panel).

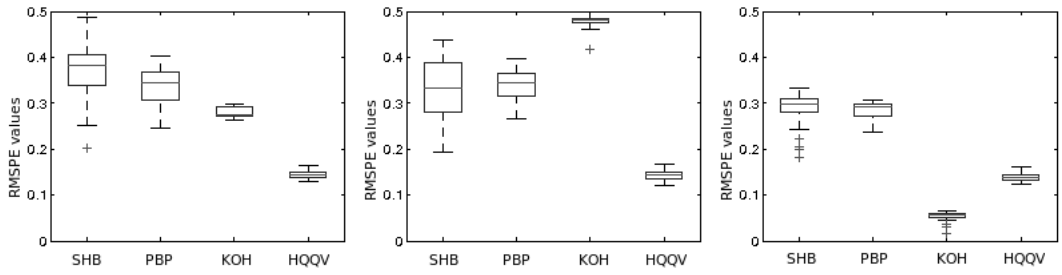

Figure 2.5: Boxplots of the 30 extrapolation RMSPEs of $\widehat{y}^{SHB}(\cdot)$, $\widehat{y}^{PBP}(\cdot)$, $\widehat{y}^{KOH}(\cdot)$, and $\widehat{y}^{HQV}(\cdot)$. The three panels correspond to Process 1 (the left panel), Process 2 (the middle panel), and Process 3 (the right panel).

To help interpret the figures, we first identify three types of information: the information in the prior (*Prior Information*), the information in the training data

taken at the level of the qualitative input where the prediction is desired (*Prediction Level Information*), and the information in the data at the remaining levels of the qualitative input (*Non-prediction Level Information*).

Figures 2.4 and 2.5 show that for both interpolation and extrapolation, $\widehat{y}^{HQV}(3,\cdot)$ has smaller prediction errors than $\widehat{y}^{SHB}(\cdot)$ for all the three processes. In particular, we have found that for the three processes, $\widehat{y}^{HQV}(3,x)$ has no larger predictive uncertainties than $\widehat{y}^{SHB}(3,x)$ for all $x \in \{0, 0.01, \ldots, 1\}$. (A measure of the predictive uncertainty has been derived in (2.9).) For example, for one of the 30 data sets generated using Process 1, the predictive uncertainty of $\widehat{y}^{HQV}(3,x)$ is about 0.55 at $x = 0.00$, but the predictive uncertainty of $\widehat{y}^{SHB}(3,x)$ is about 0.73 at $x = 0.00$. Our intuition is that $\widehat{y}^{HQV}(\cdot)$ is able to use information from all levels of the qualitative input so that there are smaller uncertainties in the predictions. (We did not compare the predictive uncertainties of $\widehat{y}^{HQV}(\cdot)$ and $\widehat{y}^{SHB}(\cdot)$ with the predictive uncertainties of $\widehat{y}^{PBP}(\cdot)$ and $\widehat{y}^{KOH}(\cdot)$ because the variance estimate of the predictive uncertainty used by frequentist predictors are not comparable with the posterior variance measure used by Bayesian hierarchical models.)

The predictor $\widehat{y}^{KOH}(\cdot)$ uses both Prediction Level Information and Non-prediction Level Information. Figure 2.4 shows that when used for interpolation, $\widehat{y}^{KOH}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ are comparable and both have smaller RMSPE than the other two predictors. Figure 2.5 shows that when used for extrapolation, if the outputs at different levels of the qualitative input are nearly parallel, $\widehat{y}^{KOH}(\cdot)$ can capture this common shape and have better extrapolation accuracy than the other three predictors (the right panel in Figure 2.5). But if the shapes of the curves differ substantially, $\widehat{y}^{KOH}(\cdot)$ can be worse than any of the other predictors (the middle panel in Figure 2.5). However, $\widehat{y}^{HQV}(\cdot)$

performs well even if the three curves are not parallel and is thus more robust (e.g., the left panel and the middle panel in Figure 2.5).

From Figures 2.4 and 2.5, $\widehat{y}^{HQV}(\cdot)$ has smaller RMSPEs than $\widehat{y}^{PBP}(\cdot)$. Our intuition is that, in this example, $\widehat{y}^{PBP}(\cdot)$ would do better if it could borrow more information from the data at the levels $t = 1$ and $t = 2$. Combining the model for $\widehat{y}^{PBP}(\cdot)$ with a Bayesian analysis or a different correlation structure might improve its performance.

In conclusion, $\widehat{y}^{HQV}(\cdot)$ is able to make effective use of the prior knowledge and the information from the training data at all the levels. It has smaller prediction errors no matter whether or not the three curves are parallel.

2.5 An Application of the Multivariate HQQV Model in Biomechanical Engineering

We apply the multivariate HQQV model to a computer code described in Rawlinson et al. (2006). This code emulated the anterior posterior displacement (APD) of the femoral component of a knee prosthesis relative to the tibial tray when the knee was loaded with a force pattern that mimics gait. The output we analyze is the APD (in millimeters) at the point that is the 13% of the way through the gait cycle which roughly corresponded to the point of the peak load and thus intuitively corresponded to the largest APD.

We consider the output as a function of four inputs. Among the four inputs, two are quantitative inputs: the *Initial Position* (IP) of the femoral component with respect to the tibial tray and the *Interface Friction* (IF) between the bone and the prosthesis. In our analysis, IP and IF are scaled to $[0, 1]^d$. Two are qualitative inputs: the *Prosthesis Design* (with value *CR* or *PS* where *CR* is a "cruciate retaining" design

that is used if the cruciate ligament is retained while *PS* is a "posterior stabilized" design that is used if the cruciate ligament is resected) and the *Loading Pattern* (with value *NG* or *SC* where *NG* is the loading corresponding to normal gait while *SC* is the loading corresponding to stair climbing). We expect that the response surfaces of the same design (or loading pattern) share more similarities than the surfaces of different designs (or loading patterns) do.

Setting $K = 2$ and $(T_1, T_2) = (2, 2)$, we apply the multivariate HQQV model with a training data set having 22 computer runs (8, 3, 7, and 4 runs for the four combinations of the qualitative inputs) to predict the outputs over a grid of the two quantitative inputs. Specifically, for each combination of the design by the loading pattern we predict the outputs on a 10 by 10 grid (obtained by crossing $\{0.05, \ldots, 0.95\}$ with itself) of the IP by the IF values. The four predicted surfaces are shown in Figure 2.6.

From the predicted surfaces we can see that (1) APD is strictly positive under SC loading but APD can be either positive or negative under NG loading, (2) APD is more sensitive to the loading pattern than to the prosthesis design; response surfaces have significantly different shapes under NG and SC loadings but similar shapes under a same loading, (3) APD is relatively insensitive to the IF values, and (4) APD increases with the IP except for IP close to 0 under SC loading (however accounting for the uncertainty in the prediction also makes it feasible that APD increases with the IP for this case). The above results are consistent with the design objectives of the CR and the PS knees.

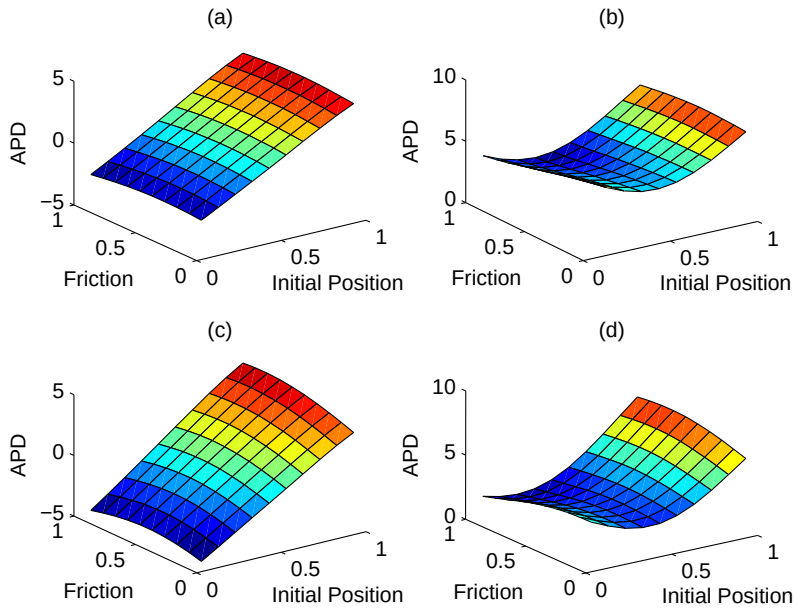

Figure 2.6: Predicted APD surfaces for the four combinations of (Prosthesis Design, Loading Pattern). The four combinations are (a) *(CR,NG)*: the upper left panel, (b) *(CR,SC)*: the upper right panel, (c) *(PS,NG)*: the lower left panel, and (d) *(PS,SC)*: the lower right panel. The two quantitative inputs are the Initial Position and the Interface Friction.

2.6 Summary and Future Research

In the examples we have presented, the HQQV model performs well for cases where the responses have similar curvatures at different levels of the qualitative input variable. At least three characteristics of the HQQV model are worth noting. First, the parameters in the HQQV model can indicate the sensitivity of the output to both quantitative inputs and the qualitative inputs. For quantitative inputs, one can conduct the sensitivity analysis by investigating the magnitude of the correlation parameters as described in Gattiker (2005) and Linkletter, Bingham, Hengartner, Higdon and Ye (2006). For the qualitative inputs, one can examine the effects of the

different levels by comparing the histograms of the posterior draws of β, σ^2, and ρ at these levels. Second, the HQQV model has $(d+2) \times \sum_{k=1}^{K} T_k$ parameters. These parameters are bounded by the hyper-priors. When there are a large number of quantitative inputs and a large number of qualitative inputs with high levels, there is the potential for the HQQV (or any) model to be nearly unidentifiable. One method of minimizing this problem is to use as strong an informative prior as one can elicit from expert opinions. Another possible method is to impose priors on the current hyper-parameters (i.e., ρ_{tj} in the HQQV model) so that with this new stage, there can be fewer parameters to be estimated and thus the non-identifiability problem could be minimized. Third, the HQQV model can adapt to both nominal and ordinal qualitative inputs. But because the qualitative variables inherently bear no order in the statistical analysis, we suggest that one should transform the ordinal variables to a quantitative scale. A method for such a transformation has been proposed by Qian et al. (2008).

Based on the HQQV model, there are several directions for future work. We classify the future research into two areas. The first area includes various applications of the HQQV model to other statistical research topics in the study of computer experiments. These topics at least include the design, optimization, calibration, and validation of computer experiments having quantitative and qualitative input variables. The second area consists of generalizations of the HQQV model. Here we note three possible generalizations. The first generalization is to model computer experiments with multivariate (or functional) output and quantitative/qualitative inputs. The second generalization is to build a stochastic model combining observations from both a physical experiment and its computer simulation code where both the physical

and the computer experiments have quantitative and qualitative inputs. The third generalization is to estimate the deterministic trend of the output from a computer code. To capture the trend, one can combine the HQQV model with at least two approaches. One approach is the blind kriging proposed by Joseph, Hung and Sudjianto (2007). The other approach allowing the HQQV model to capture the common trend (of the responses) related by different qualitative input levels is currently being developed by the authors.

Finally, we shall emphasize that the (multivariate) HQQV model is constructed based on the assumption that the responses at different levels of the qualitative input(s) are similar in terms of the correlation structures. If, in applications, this assumption is doubtful (e.g., expert knowledge suggests that the response surface is smooth at one level of t but is bumpy at another level), one can either use the HQQV model with a joint prior distribution (of $\boldsymbol{\beta}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\rho}$) determined by expert knowledge or use other plausible models, e.g., the cumulative roughness model proposed by Kennedy and O'Hagan (2000).

CHAPTER 3

ANOVA KRIGING: A METHODOLOGY FOR PREDICTING THE OUTPUT FROM A COMPLEX COMPUTER CODE HAVING QUANTITATIVE AND QUALITATIVE INPUTS

3.1 Introduction

Complex computer simulations (computer experiments) have increasing usage in the recent years. This is because, first, many physical experiments can be difficult or impossible to run, and second, mathematical equations and the computer codes implementing them are developed to mimic these physical experiments (See the chapter 1 in Santner et al. (2003) and chapter 1 in Fang et al. (2005) for examples of physical experiments and their complex computer codes).

The number of observations from a computer code are often limited because one computer experiment can typically take days to finish. The accurate prediction for a computer code with a limited number of runs is thus necessary. Statistical models have therefore been developed to predict the output from computer experiments as well as the corresponding physical experiment. Whereas Gaussian stochastic process models (Sacks et al. (1989b)) and kriging theory (Matheron (1963) and Cressie (1993)) have been well developed for predicting computer experiments having quantitative

inputs, there are fewer attempts for modeling the output from a computer code having quantitative and qualitative mixed inputs. We can group the studies for predicting the output from computer experiments having mixed inputs in terms of their focus. McMillan et al. (1999) and Qian et al. (2008) focused on the correlation function and modeled the correlation between responses at different levels of the qualitative input. Kennedy and O'Hagan (2001) and Qian and Wu (2008) focused on building stochastic processes to describe the observations and regarded the responses at all the levels as coming from linear combinations of independent Gaussian stochastic processes. In Chapter 2, we used Bayesian analysis with an empirical prior to capture the similarities of the responses at different levels.

The above works have at least two limitations. First, none of them provided condition(s) under which a predictor would have a smaller prediction error than the ordinary kriging predictor using the data at one level of the qualitative input. Second, none addressed the issue as to whether combining the data in all the levels can help improve the prediction accuracy. As will be shown in Section 3, including the training data at a level where the responses differ substantially from the responses at other levels can increase the prediction error.

This chapter proposes a methodology for predicting the output from computer experiments having quantitative and qualitative mixed inputs. Our predictor can capture the common trend of the responses. The methodology we propose differs from the previously proposed methods in that if the responses at level $t^\star$ have a substantially different shape comparing with the responses at other levels, our predictor

will not use the observations at level $t^\star$. Thus, the prediction for a computer experiment will not be affected by invalid (or incorrectly calibrated/tuned) computer codes.

The outline of this chapter is as follows: Section 3.2 introduces our predictor for computer codes having one qualitative input. Section 3.3 generalizes our predictor to the prediction for computer experiments having multiple qualitative inputs. Section 3.4 illustrates this method in a biomechanical engineering application. A summary of Chapter 3 is given in Section 5.

3.2 The ANOVA Kriging Model for Computer Experiments Having One Qualitative Input

This section proposes the HQQV ANOVA kriging (HAK) predictor. Section 3.2.1-3.2.2 compare the ordinary predictor with an average effect predictor. Section 3.2.3 and Section 3.2.4 propose an ANOVA kriging predictor in and the HAK predictor. Section 3.2.5 illustrates the HAK predictor in a numerical example. In Section 3.2.1–3.2.3, the model views the computer experiment output as a draw from a second-order stationary stochastic process. Section 3.2.4–3.2.5 combine the kriging predictors with a Gaussian stochastic process model.

3.2.1 Ordinary Kriging and Average Effect Kriging

Kriging is desirable for predicting the output from a computer code because kriging predictors are interpolators and kriging needs no assumption about the probabilistic distribution of the process model (Cressie (1993), chapter 3). If the trend of the unknown response function is treated as an unknown constant, the kriging predictor is known as the ordinary kriging predictor. Next, the ordinary kriging predictor is

constructed as well as an "average effect" kriging predictor for computer experiments having one quantitative variable with T levels. We let $t \in \{1, \ldots, T\}$ denote the qualitative input and $\boldsymbol{x}$ denote the quantitative input in $[0,1]^d$ (scaled if necessary). Let $y(t, \boldsymbol{x})$ denote the output at $(t, \boldsymbol{x})$ and $Y(t, \boldsymbol{x})$ denote the corresponding random function. Conditionally, assume the second order stationary condition so that $E(Y(t, \boldsymbol{x})) = \beta_t$ for all $\boldsymbol{x} \in [0,1]^d$ and $Cov(Y(t, \boldsymbol{x}_1), Y(t, \boldsymbol{x}_2)) = C(\boldsymbol{x}_1 - \boldsymbol{x}_2 | \boldsymbol{\theta}_t)$ for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in [0,1]^d$, where β_t is an unknown constant, $\boldsymbol{\theta}_t$ is the vector of correlation parameters, and $C(\cdot|\boldsymbol{\theta})$ is the covariance function (covariogram). Further, assume that $Y(t_1, \boldsymbol{x})$ and $Y(t_2, \boldsymbol{x})$ are independent if $t_1 \neq t_2$ for all $t_1, t_2 \in \{1, \ldots, T\}$.

For all $t \in \{1, \ldots, T\}$, let n_t denote the number of inputs at level t and $\boldsymbol{x}_1^{(t)}, \ldots, \boldsymbol{x}_{n_t}^{(t)}$ denote the inputs at level t. Let

$$\boldsymbol{y}^{(t)} = (y(t, \boldsymbol{x}_1^{(t)}), \ldots, y(t, \boldsymbol{x}_{n_t}^{(t)}))^\top$$

denote the vector of the responses at $(t, \boldsymbol{x}_1^{(t)}), \ldots, (t, \boldsymbol{x}_{n_t}^{(t)})$ at level t and regard $\boldsymbol{y}^{(t)}$ as a realization of the random vector

$$\boldsymbol{Y}^{(t)} = (Y(t, \boldsymbol{x}_1^{(t)}), \ldots, Y(t, \boldsymbol{x}_{n_t}^{(t)}))^\top.$$

Let $y(t_0, \boldsymbol{x}_0)$ denote an unknown output. The ordinary kriging predictor of $y(t_0, \boldsymbol{x}_0)$ based on the data observed at level t_0 is $\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0^{(t_0)})^\top \boldsymbol{y}^{(t_0)}$, where $\boldsymbol{y}^{(t_0)}$ are observations at level t_0 and $\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0) = \left(\lambda_1^{(t_0, \boldsymbol{x}_0)}, \ldots, \lambda_{n_{t_0}}^{(t_0, \boldsymbol{x}_0)}\right)^\top$ satisfies $\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \times \mathbf{1}_{n_{t_0} \times 1}$ $= \sum_{i=1}^{n_{t_0}} \lambda_i^{(t_0, \boldsymbol{x}_0)} = 1$ and $\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0^{t_0})^\top \boldsymbol{y}^{(t_0)}$ minimizes the mean squared prediction error among all the linear predictors; i.e.,

$$\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}}\, E(Y(t_0, \boldsymbol{x}_0) - \boldsymbol{\lambda}^\top \boldsymbol{y}^{(t_0)})^2, \tag{3.1}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{n_{t_0}})^\top$ is any n_{t_0} by 1 vector with $\boldsymbol{\lambda}^\top \mathbf{1}_{n_{t_0} \times 1} = 1$. According to the theory of kriging (Matheron (1963)), $\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)$ can be written as a function

of the covariogram $C(\cdot|\boldsymbol{\theta}_{t_0})$ under the second order stationarity. Let $\widehat{y}^{OK}(t_0, \boldsymbol{x}_0) = \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t_0)}$ denote the ordinary kriging predictor of $y(t_0, \boldsymbol{x}_0)$.

Next, we introduce a predictor named *the average effect kriging predictor*. The idea is that we predict the trend $A(\cdot)$ at $\boldsymbol{x}$

$$A(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} y(t, \boldsymbol{x})$$

and the deviation from the trend $y(t_0, \boldsymbol{x}_0) - A(\boldsymbol{x}_0)$; the sum of these two predictors is the average-effect kriging predictor of $y(t_0, \boldsymbol{x}_0)$. Specifically, with the second order stationarity,

$$\widehat{A}(\boldsymbol{x}_0) = \sum_{t=1}^{T} \frac{1}{T}\boldsymbol{\lambda}(t, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t)} \tag{3.2}$$

is a predictor of $A(\boldsymbol{x}_0) = \frac{1}{T}\sum_{t=1}^{T} y(t, \boldsymbol{x})$.

By subtracting $\widehat{A}(\cdot)$ from the training data we obtain a new set of observations

$$\begin{aligned} \boldsymbol{y}^{\star(t_0)} &= (y^\star(t_0, \boldsymbol{x}_1^{(t_0)}), \ldots, y^\star(t_0, \boldsymbol{x}_{n_{t_0}}^{(t_0)}))^\top \\ &= \left(y(t_0, \boldsymbol{x}_1^{(t_0)}) - \widehat{A}(\boldsymbol{x}_1^{(t_0)}), \ldots, y(t_0, \boldsymbol{x}_{n_{t_0}}^{(t_0)}) - \widehat{A}(\boldsymbol{x}_{n_{t_0}}^{(t_0)})\right)^\top \end{aligned} \tag{3.3}$$

with the corresponding random vector being

$$\boldsymbol{Y}^{\star(t_0)} = \left(Y(t_0, \boldsymbol{x}_1^{(t_0)}) - \widehat{A}(\boldsymbol{x}_1^{(t_0)}), \ldots, Y(t_0, \boldsymbol{x}_{n_{t_0}}^{(t_0)}) - \widehat{A}(\boldsymbol{x}_{n_{t_0}}^{(t_0)})\right)^\top. \tag{3.4}$$

We regard $\boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{\star(t_0)}$ as a predictor of $(y(t_0, \boldsymbol{x}_0) - A(\boldsymbol{x}_0))$ where $\boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0)$ is an $n_{t_0} \times 1$ vector such that $\boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0) \times \mathbf{1} = 1$ and $\boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{\star(t_0)}$ is an interpolator of the data points in $\boldsymbol{y}^{\star(t_0)}$. We regard the average effect predictor $\widehat{\boldsymbol{y}}^{AE}(t_0, \boldsymbol{x}_0)$ as the

sum of $\widehat{A}(\boldsymbol{x}_0)$ and $\boldsymbol{\lambda}^\star(t_0,\boldsymbol{x}_0)^\top \boldsymbol{y}^{\star(t_0)}$; i.e.,

$$\begin{aligned}
\widehat{y}^{AE}(t_0,\boldsymbol{x}_0) &= \boldsymbol{\lambda}^\star(t_0,\boldsymbol{x}_0)^\top \boldsymbol{y}^{\star(t_0)} + \widehat{A}(\boldsymbol{x}_0) \\
&= \boldsymbol{\lambda}^\star(t_0,\boldsymbol{x}_0)^\top \left(\boldsymbol{y}^{(t_0)} - \begin{pmatrix} \sum_{t=1}^T \frac{1}{T}\boldsymbol{\lambda}(t,\boldsymbol{x}_1^{(t_0)})^\top \boldsymbol{y}^{(t)} \\ \vdots \\ \sum_{t=1}^T \frac{1}{T}\boldsymbol{\lambda}(t,\boldsymbol{x}_{n_{t_0}}^{(t_0)})^\top \boldsymbol{y}^{(t)} \end{pmatrix} \right) \\
&\quad + \sum_{t=1}^T \frac{1}{T}\boldsymbol{\lambda}(t,\boldsymbol{x}_0)^\top \boldsymbol{y}^{(t)}. \qquad (3.5)
\end{aligned}$$

It can be shown that both $\widehat{y}^{OK}(\cdot)$ and $\widehat{y}^{AE}(\cdot)$ are unbiased interpolators. The unbiasedness and the interpolating property of $\widehat{y}^{OK}(\cdot)$ can be found in Matheron (1963) and Cressie (1993), chapter 3. To show that $\widehat{y}^{AE}$ is unbiased, we use the unbiased property of the $\widehat{y}^{OK}$ (which is $E(\widehat{y}^{OK}(t,\boldsymbol{x}_0)) = \beta_t$); i.e.,

$$\begin{aligned}
E(\widehat{y}^{AE}(t_0,\boldsymbol{x}_0)) &= \boldsymbol{\lambda}^\star_{(t_0,\boldsymbol{x}_0)} \times \left[E(\boldsymbol{Y}^{(t_0)} - \begin{pmatrix} \sum_{t=1}^T \frac{1}{T}\boldsymbol{\lambda}^\top_{(t,\boldsymbol{x}_1^{(t_0)})} E(\boldsymbol{Y}^{(t)}) \\ \vdots \\ \sum_{t=1}^T \frac{1}{T}\boldsymbol{\lambda}^\top_{(t,\boldsymbol{x}_{n_{t_0}}^{(t_0)})} E(\boldsymbol{Y}^{(t)}) \end{pmatrix} \right] \\
&\quad + \sum_{t=1}^T \frac{1}{T}\boldsymbol{\lambda}^\top_{(t_0,\boldsymbol{x}_0)} E(\boldsymbol{Y}^{(t)}) \\
&= \boldsymbol{\lambda}^\star(t_0,\boldsymbol{x}_0)^\top (\beta_{t_0} \times \mathbf{1} - \sum_{t=1}^T \frac{1}{T}\beta_t \times \mathbf{1}) + \sum_{t=1}^T \frac{1}{T}\boldsymbol{\beta}_t \\
&= \beta_{t_0} = E(Y(t_0,\boldsymbol{x}_0)).
\end{aligned}$$

To show that $\widehat{y}^{AE}$ is an interpolator, we use the result that $\lambda^\star(t_0,\boldsymbol{x}_0)\boldsymbol{y}^{(t_0)}$ is an interpolator so that $\widehat{y}^{AD}(t_0,\boldsymbol{x}_0) = \lambda^\star(t_0,\boldsymbol{x}_0)^\top \boldsymbol{y}^\star(t_0) + \widehat{A}(\boldsymbol{x}_0) = y(t_0,\boldsymbol{x}_0) - \widehat{A}(\boldsymbol{x}_0) + \widehat{A}(\boldsymbol{x}_0) = y(t_0,\boldsymbol{x}_0)$ is an interpolator.

3.2.2 Comparing $\widehat{y}^{OK}(\cdot)$ and $\widehat{y}^{AE}(\cdot)$

We compare the predictive accuracies of $\widehat{y}^{OK}$ and $\widehat{y}^{AE}$ for predicting $y(t_0,\boldsymbol{x}_0)$ here and propose our predictor in Section 3.2.3. A measure to describe the prediction error

of the predictor $\widehat{y}(t_0, \boldsymbol{x}_0)$ is the (mean) squared prediction error $V(t_0, \boldsymbol{x}_0)$; i.e.,

$$V(t_0, \boldsymbol{x}_0) = E(\widehat{y}(t_0, \boldsymbol{x}_0) - y(t_0, \boldsymbol{x}_0))^2 = (\widehat{y}(t_0, \boldsymbol{x}_0) - y(t_0, \boldsymbol{x}_0))^2. \tag{3.6}$$

The second equation in (3.6) holds because here we regard $y(\cdot)$ as an unknown deterministic function of $(t, \boldsymbol{x})$. We let $V^{OK}(t_0, \boldsymbol{x}_0)$ and $V^{AE}(t_0, \boldsymbol{x}_0)$ denote the mean squared prediction errors of $\widehat{y}^{OK}(t_0, \boldsymbol{x}_0)$ and $\widehat{y}^{AE}(t_0, \boldsymbol{x}_0)$. Proposition 1 quantifies the discrepancy between the squared errors of the two predictors.

Proposition 1. If

1. let $\boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0) = \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)$, where $\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)$ and $\boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0)$ are in (3.1) and (3.5), respectively,

2. the *quantitative* inputs at level t_0 $\{\boldsymbol{x}_1^{(t_0)}, \ldots, \boldsymbol{x}_{n_{t_0}}^{(t_0)}\}$ and $\boldsymbol{x}_0$ are quantitative inputs at level t for all $t \neq t_0$; i.e., $\{(t, \boldsymbol{x}_1^{(t_0)}), \ldots, (t, \boldsymbol{x}_{n_{t_0}}^{(t_0)}), (t, \boldsymbol{x}_0)\}$ are inputs to the computer code for all $t \neq t_0$,

then

$$V^{OK}(t_0, \boldsymbol{x}_0) - V^{AE}(t_0, \boldsymbol{x}_0) = \frac{2}{T} B \times D - \frac{1}{T^2} D^2, \tag{3.7}$$

where

$$B = \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t_0)} - y(t_0, \boldsymbol{x}_0) \tag{3.8}$$

denotes the deviation from the predictor at t_0 and

$$D = \sum_{t \neq t_0} \left[\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \begin{pmatrix} y(t, \boldsymbol{x}_1^{(t_0)}) \\ \vdots \\ y(t, \boldsymbol{x}_{n_{t_0}}^{(t_0)}) \end{pmatrix} - y(t, \boldsymbol{x}_0) \right] \tag{3.9}$$

denotes the deviations from the predictors at levels other than t_0.

Proof. Notice that $V^{OK}(t_0, \boldsymbol{x}_0)$ can be derived as

$$V^{OK}(t_0, \boldsymbol{x}_0) = \left(\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t_0)} - y(t_0, \boldsymbol{x}_0) \right)^2 = B^2. \tag{3.10}$$

On the other hand, $\widehat{y}^{AE}$ can be written as

$$\begin{aligned}
\widehat{y}^{AE}(t_0, \boldsymbol{x}_0) &= \boldsymbol{\lambda}^{\star}(t_0, \boldsymbol{x}_0)^{\top} \left(\boldsymbol{y}^{(t_0)} - \begin{pmatrix} \sum_{t=1}^{T} \frac{1}{T} \boldsymbol{\lambda}(t, \boldsymbol{x}_1^{(t_0)})^{\top} \boldsymbol{y}^{(t)} \\ \vdots \\ \sum_{t=1}^{T} \frac{1}{T} \boldsymbol{\lambda}(t, \boldsymbol{x}_{n_{t_0}}^{(t_0)})^{\top} \boldsymbol{y}^{(t)} \end{pmatrix} \right) \\
&\quad + \sum_{t=1}^{T} \frac{1}{T} \boldsymbol{\lambda}(t, \boldsymbol{x}_0)^{\top} \boldsymbol{y}^{(t)} \\
&= \boldsymbol{\lambda}^{\star}(t_0, \boldsymbol{x}_0)^{\top} \left(\frac{T-1}{T} \boldsymbol{y}^{(t_0)} - \begin{pmatrix} \sum_{t \neq T} \frac{1}{T} y(t, \boldsymbol{x}_1^{(t_0)}) \\ \vdots \\ \sum_{t=1}^{T} \frac{1}{T} y(t, \boldsymbol{x}_{n_{t_0}}^{(t_0)}) \end{pmatrix} \right) \\
&\quad + \sum_{t \neq T} \frac{1}{T} y(t, \boldsymbol{x}_0) + \frac{1}{T} \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^{\top} \boldsymbol{y}^{(t_0)} \\
&= \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^{\top} \boldsymbol{y}^{(t_0)} - \frac{1}{T} D = \widehat{y}^{OK}(t_0, \boldsymbol{x}_0) - \frac{1}{T} D. \qquad (3.11)
\end{aligned}$$

So

$$V^{AE}(t_0, \boldsymbol{x}_0) = \left(\widehat{y}^{OK}(t_0, \boldsymbol{x}_0) - y(t_0, \boldsymbol{x}_0) - \frac{1}{T} D \right)^2 = \left(B - \frac{1}{T} D \right)^2,$$

and

$$V^{OK}(t_0, \boldsymbol{x}_0) - V^{AE}(t_0, \boldsymbol{x}_0) = B^2 - (B - \frac{1}{T} D)^2 = \frac{2}{T} BD - \frac{1}{T^2} D^2. \quad \blacksquare$$

Proposition 2 provides conditions guaranteeing that $\widehat{y}^{AE}$ has no larger squared prediction error than $\widehat{y}^{OK}$.

Proposition 2. If the responses at different levels of the qualitative input have the same shape; i.e.,

$$y(t, \boldsymbol{x}) = \beta_t + B(\boldsymbol{x}),$$

where β_t is the unknown mean and $B(\boldsymbol{x})$ is the unknown common trend centered at 0 so that $\int B(\boldsymbol{x}) d\boldsymbol{x} = 0$, then with the two conditions in Proposition 1,

$$V^{OK}(t_0, \boldsymbol{x}_0) - V^{AE}(t_0, \boldsymbol{x}_0) = \frac{T^2 - 1}{T^2} \left(\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^{\top} \boldsymbol{B}^{(t_0)} - B(\boldsymbol{x}_0) \right)^2 \geq 0, \qquad (3.12)$$

where $\boldsymbol{B}^{(t)}$ is an $n_t \times 1$ vector whose ith element is $B(\boldsymbol{x}_i^{(t_0)})$ for all $i = 1, \ldots, n_t$.

Proof. Notice that $V^{OK}(t_0, \boldsymbol{x}_0)$ can be derived as

$$\begin{aligned} V^{OK}(t_0, \boldsymbol{x}_0) &= \left(\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t_0)} - y(t_0, \boldsymbol{x}_0)\right)^2 \\ &= \left(\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top (\beta_{t_0} \times \mathbf{1}_{n_{t_0} \times 1} + \boldsymbol{B}^{(t_0)}) - (\beta_{t_0} + B(\boldsymbol{x}_0))\right)^2 \\ &= \left(\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{B}^{(t_0)} - B(\boldsymbol{x}_0)\right)^2. \end{aligned} \tag{3.13}$$

Because $\left\{\boldsymbol{x}_1^{(t_0)}, \ldots, \boldsymbol{x}_{n_{t_0}}^{(t_0)}, \boldsymbol{x}_0\right\}$ are inputs to the level $t \neq t_0$,

$$\begin{aligned} \widehat{y}^{AE}(t_0, \boldsymbol{x}_0) &= \left[\frac{T-1}{T}\boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0)^\top + \frac{1}{T}\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top\right] \times (\beta_{t_0} \times \mathbf{1}_{n_{t_0}} + \boldsymbol{B}^{(t_0)}) \\ &\quad + \frac{T-1}{T}B(\boldsymbol{x}_0) + \sum_{t \neq t_0} \frac{\beta_t}{T} \\ &\quad - \boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0)^\top \left[\left(\sum_{t \neq t_0} \frac{\beta_t}{T}\right) \times \mathbf{1}_{n_{t_0} \times 1} + \frac{T-1}{T}\boldsymbol{B}^{(t_0)}\right] \\ &= \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top (\beta_{t_0} \times \mathbf{1}_{n_{t_0} \times 1} + \boldsymbol{B}^{(t_0)}) - \frac{T-1}{T}\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{B}^{(t_0)} \\ &\quad + \frac{T-1}{T}B(\boldsymbol{x}_0) \\ &= \beta_{t_0} + \frac{1}{T}\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{B}^{(t_0)} + \frac{T-1}{T}B(\boldsymbol{x}_0). \end{aligned} \tag{3.14}$$

By (3.14),

$$\begin{aligned} V^{AE}(t_0, \boldsymbol{x}_0) &= \left[\beta_{t_0} + \frac{1}{T}\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{B}^{(t_0)} + \frac{T-1}{T}B(\boldsymbol{x}_0) - (\beta_{t_0} + B(\boldsymbol{x}_0))\right]^2 \\ &= \left(\frac{1}{T}\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)\boldsymbol{B}^{(t_0)} - \frac{1}{T}B(\boldsymbol{x}_0)\right)^2 \\ &= \frac{1}{T^2}\left(\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{B}^{(t_0)} - B(\boldsymbol{x}_0)\right)^2. \end{aligned} \tag{3.15}$$

By Equations 3.13 and 3.15,

$$V^{OK}(t_0, \boldsymbol{x}_0) - V^{AE}(t_0, \boldsymbol{x}_0) = \frac{T^2-1}{T^2}\left(\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{B}^{(t_0)} - B(\boldsymbol{x}_0)\right)^2. \quad \blacksquare$$

Proposition 2 implies that if the responses at different levels share a common trend, using $\widehat{y}^{AE}$ results in a no bigger prediction error than $\widehat{y}^{OK}$. In reality, however,

one usually has no information about the trend of the responses. We thus propose a new predictor with a cross-validation procedure to select the levels to be included for the prediction.

3.2.3 The ANOVA Kriging Predictor for Computer Experiments Having an Arbitrary Number of Quantitative Inputs and One Qualitative Input

For computer experiments having $d \geq 1$ dimensional quantitative input $\boldsymbol{x} \in [0,1]^d$ and one qualitative input $t \in \{1, \ldots, T\}$, we construct the ANOVA kriging predictor of $y(t_0, \boldsymbol{x}_0)$ as follows. **Step 1**-**Step 4** select levels to estimate the average effect; **Step 5**-**Step 7** use these levels to predict $y(t_0, \boldsymbol{x}_0)$.

Step 1 For all $t \neq t_0$, augment the training data at level t by predicting $(t, \boldsymbol{x}_1^{(t_0)}), \ldots, (t, \boldsymbol{x}_{n_{t_0}}^{(t_0)}), (t, \boldsymbol{x}_0)$ and merging these predictions with the training data set.

Step 2 Predict the T responses $y(1, \boldsymbol{x}_0), \ldots, y(T, \boldsymbol{x}_0)$. Let $\widehat{y}(t_0, \boldsymbol{x}_0) = \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t_0)}$ denote the predictor of $y(t_0, \boldsymbol{x}_0)$.

Step 3 Construct all 2^{T-1} combinations of the levels $1, \ldots, t_0 - 1, t_0 + 1, \ldots, T$.

Step 4 For each of the combinations, suppose $t_1^c, \ldots, t_{P_c}^c$ are the levels in the selected combination. Compute

$$\sum_{i=1}^{n_{t_0}} \left\{ \frac{2}{T} B_{-i} D_{-i} - \frac{1}{T^2} D_{-i}^2 \right\}, \tag{3.16}$$

where B_{-i} and D_{-i} are computed using (3.8) and (3.9) based on the data set

$\Big\{ y(t_0, \boldsymbol{x}_1^{(t_0)}), \ldots, y(t_0, \boldsymbol{x}_{i-1}^{(t_0)}), y(t_0, \boldsymbol{x}_{i+1}^{(t_0)}), \ldots, y(t_0, \boldsymbol{x}_{n_{t_0}}^{(t_0)}),$
$\widehat{y}(t_1^c, \boldsymbol{x}_1^{(t_0)}), \ldots, \widehat{y}(t_{P_c}^c, \boldsymbol{x}_1^{(t_0)}), \ldots, \widehat{y}(t_1^c, \boldsymbol{x}_{i-1}^{(t_0)}), \ldots, \widehat{y}(t_{P_c}^c, \boldsymbol{x}_{i-1}^{(t_0)}),$
$\widehat{y}(t_1^c, \boldsymbol{x}_{i+1}^{(t_0)}), \ldots, \widehat{y}(t_{P_c}^c, \boldsymbol{x}_{i+1}^{(t_0)}), \ldots, \widehat{y}(t_1^c, \boldsymbol{x}_{n_{t_0}}^{(t_0)}), \ldots, \widehat{y}(t_{P_c}^c, \boldsymbol{x}_{n_{t_0}}^{(t_0)}) \Big\}$, which is the augmented training data set without $y(t_0, \boldsymbol{x}_i^{(t_0)}), y(t_1^c, \boldsymbol{x}_i^{(t_0)}), y(t_1^c, \boldsymbol{x}_i^{(t_0)}), \ldots, y(t_{P_c}^c, \boldsymbol{x}_i^{(t_0)})$.

Therefore, the AK predictor predictor using levels $t_1^c, \ldots, t_{P_c}^c$ to estimate the average effect has the smallest cross-validated prediction error for the training data at level t_0 $y(t_0, \boldsymbol{x}_1^{(t_0)}), \ldots, y(t_0, \boldsymbol{x}_{n_{t_0}}^{(t_0)})$.

Step 5 Denote the set of levels corresponding to the smallest $\sum_{i=1}^{n_{t_0}} \{\frac{2}{T} B_{-i} D_{-i} - \frac{1}{T^2} D_{-i}^2\}$ as $t_1^{AK}, \ldots, t_P^{AK}$. Compute $\widehat{A}^{AK}(\boldsymbol{x}_1^{(t_0)}), \ldots, \widehat{A}^{AK}(\boldsymbol{x}_{n_{t_0}}^{(t_0)})$, $\widehat{A}^{AK}(\boldsymbol{x}_0)$ using the formula

$$\widehat{A}^{AK}(\boldsymbol{x}) = \frac{1}{P+1} \left[\lambda(t_0, \boldsymbol{x})^\top \boldsymbol{y}^{(t_0)} + \sum_{t=t_1^{AK}}^{t_P^{AK}} \lambda(t, \boldsymbol{x})^\top \widehat{\boldsymbol{y}}^{(t)} \right], \tag{3.17}$$

where $\widehat{\boldsymbol{y}}^{(t)} = (\widehat{y}(t, \boldsymbol{x}_1^{(t_0)}), \ldots, \widehat{y}(t, \boldsymbol{x}_{n_{t_0}}^{(t_0)}))^\top$.

Step 6 Construct the deviation data set $\boldsymbol{y}^{\star(t_0)}$ using (3.3).

Step 7 Compute the ANOVA kriging predictor of $y(t_0, \boldsymbol{x}_0)$

$$\widehat{y}^{AK}(t_0, \boldsymbol{x}_0) = \widehat{A}^{AK}(\boldsymbol{x}_0) + \boldsymbol{\lambda}^\star(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{\star(t_0)}.$$

This procedure can be combined with any predictive model producing a predictor having the form $\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t_0)}$. We introduce a predictor based on a Gaussian process model with hyper priors next.

3.2.4 The HAK Predictor

We use the Gaussian stochastic process model that has been proposed in Chapter 2. Given parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_T)^\top$, $\boldsymbol{\sigma} = (\sigma_1^2, \ldots, \sigma_T^2)^\top$, and $\boldsymbol{\rho} = (\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_T)^\top$ where $\boldsymbol{\rho}_t = (\rho_{t,1}, \ldots, \rho_{t,d})^\top$, our GaSP model views the output $y(t, \boldsymbol{x})$ as a realization of

$$Y(t, \boldsymbol{x}) = \beta_t + Z_t(\boldsymbol{x}),$$

where $Z_t(\boldsymbol{x})$ is a stationary Gaussian stochastic process with mean zero, variance σ_t^2, and the *Gaussian correlation function*; i.e.,

$$Cov(Z_t(\boldsymbol{x}_1), Z_t(\boldsymbol{x}_2)) = \prod_{i=1}^{d} \rho_{t,i}^{(x_{1,i}-x_{2,i})^2}.$$

Further, $Z_1(\cdot), \ldots, Z_T(\cdot)$ are mutually independent Gaussian processes.

Following the above model, one can do prediction using either the frequentist or the Bayesian method. Here we use the hierarchical Bayesian predictor (the HQQV predictor) proposed in Chapter 2 so that our predictor can share the merits of both the ANOVA kriging and the Bayesian analysis in that the prediction of the response at one level can borrow information about the common shape and the correlation structure from the responses at all the levels. Thus, this chapter proposes $\widehat{y}^{HAK}(t_0, \boldsymbol{x}_0)$ as the HQQV ANOVA kriging (HAK) predictor. Next, $\widehat{y}^{HAK}(t_0, \boldsymbol{x}_0)$ is compared with two alternative predictors in a numerical example next.

3.2.5 An Example Having One Quantitative Input and One Qualitative Input Having Three Levels

Let $\widehat{y}^{KOH}(\cdot)$ and $\widehat{y}^{HQV}(\cdot)$ denote the predictors in Kennedy and O'Hagan (2001) and the HQQV predictor in Chapter 2, respectively. This section compares $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{HQV}(\cdot)$, and $\widehat{y}^{HAK}(\cdot)$ in three cases described in Chapter 2.4.3. In these cases, the true data are three quadratic curves having the form $y(1,x) = b_{01} + b_{11}x + b_{21}x^2$, $y(2,x) = b_{02} + b_{12}x + b_{22}x^2$, and $y(3,x) = b_{03} + b_{13}x + b_{23}x^2$ for $x \in [0,1]$, and the observations are $x \in \{0, 0.25, 0.5, 0.75, 1\}$ for t=1,2 and $x \in \{0.5, 0.75, 1\}$ for $t = 3$.

As studied in Section 2.4.3, $(x_1, x_2, \ldots, x_{n_{\text{pred}}}) = (0.5, 0.51, \ldots, 1.00)$ for interpolation and $(x_1, x_2, \ldots, x_{n_{\text{pred}}}) = (0, 0.01, \ldots, 0.50)$ for extrapolation. For each of the three cases, 30 true data sets were generated and the interpolation and extrapolation

RMSPEs in (2.16) were computed. The boxplots in Figures 3.1 and 3.2 display RMSPEs of the three predictors for both interpolation and extrapolation. We can see

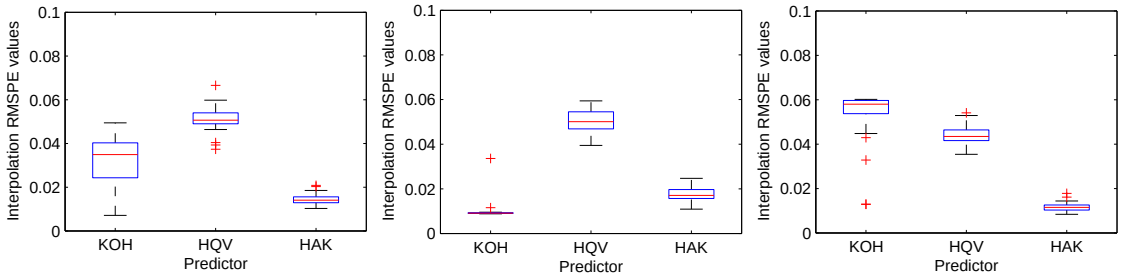

Figure 3.1: Boxplots of the 30 interpolation RMSPEs of $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{HQV}(\cdot)$, and $\widehat{y}^{HAK}(\cdot)$. The three panels correspond to Mechanism 1 (the left panel), Mechanism 2 (the middle panel), and Mechanism 3 (the right panel).

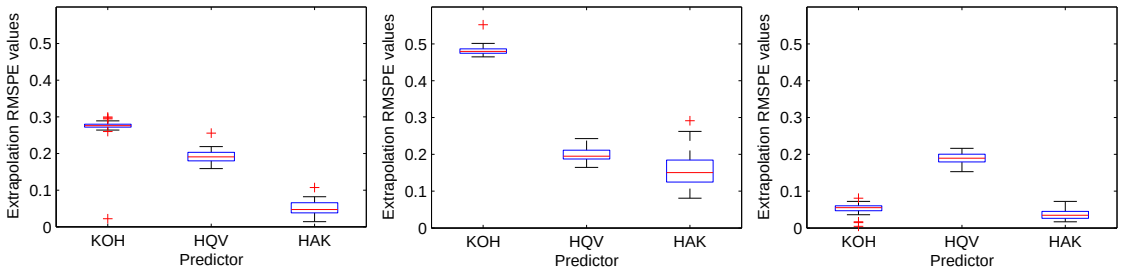

Figure 3.2: Boxplots of the 30 extrapolation RMSPEs of $\widehat{y}^{KOH}(\cdot)$, $\widehat{y}^{HQV}(\cdot)$, and $\widehat{y}^{HAK}(\cdot)$. The three panels correspond to Mechanism 1 (the left panel), Mechanism 2 (the middle panel), and Mechanism 3 (the right panel).

that $\widehat{y}^{HAK}$ generally has the smallest RMSPEs than the other two predictors for all the cases. Intuitively, the HAK predictor has the advantage of the HQQV predictor

because it builds on the HQQV model. Further, the HAK predictor generally has smaller prediction error than the HQQV predictor because it takes into account the common trend. Further, the HAK kriging predictor can borrow information of the common shapes in a data adaptive fashion so that the predictor chooses level 1 and 2 for case 1 and case 3 where the three curves have similar shapes; the predictor selects level 2 for case 2 where the curve at level 1 has a different shape comparing with the other two curves. Because of this feature, $\widehat{y}^{HAK}$ has smaller prediction than $\widehat{y}^{KOH}$ in the middle panel of Figure 3.2 where $\widehat{y}^{KOH}$ uses the data at all the levels.

3.3 The HAK Predictor for Computer Experiments Having Multiple Qualitative Inputs

In this section, we extend the current HAK predictor to computer experiments having multiple qualitative inputs. Our idea is to estimate the main/interaction effects using a subset of the training data and combining the deviations and the main/interaction effects. We introduce the HAK predictor for computer experiments having *two* qualitative inputs in Section 3.3.1. We extend the work to computer experiments having an arbitrary number of qualitative inputs in Section 3.3.2.

3.3.1 The HAK Model for Computer Experiments Having Two Qualitative Inputs

Suppose the input has the form $(t^{(1)}, t^{(2)}, \boldsymbol{x})$, where $t^{(1)} \in \{1, \ldots, T_1\}$ and $t^{(2)} \in \{1, \ldots, T_2\}$ are two qualitative inputs. The estimation of the average effect $A(\cdot)$, the main effects of $t^{(1)}$ and $t^{(2)}$, and their interaction effect are described next.

To estimate the average effect, which is the common trend of the responses at all the levels $\{(1,1), \ldots, (1,T_2), \ldots, (T_1,1), \ldots, (T_1,T_2)\}$, our method constructs a new

qualitative variable t having $T_1 \times T_2$ levels. So each level of t corresponds to a combination of the values of $t^{(1)}$ and $t^{(2)}$. Then we follow the Steps $1-7$ in Section 2.3 to compute the trend. We regard $\widehat{A}^{AK}(\boldsymbol{x})$ as the estimated average effect in the step 5 in Section 3.2.3 and $y^\star(t^{(1)}, t^{(2)}, \boldsymbol{x})$ as the deviation obtained by subtracting the average effect $\widehat{A}^{AK}(\boldsymbol{x})$ from the true response $y(t^{(1)}, t^{(2)}, \boldsymbol{x})$.

The main effect of $t^{(1)} = t_1$ is the average of the outputs at levels $(t^{(1)}, t^{(2)}) = (t_1, 1)$ and $(t^{(1)}, t^{(2)}) = (t_1, 2)$. To estimate the main effect of $t^{(1)}$, we use the data points having $t^{(1)} = t_1$ to estimate the main effect using Formula 3.2. So the main effect of the first qualitative input at t_1 is

$$\widehat{A}^{AK}(t^{(1)} = t_1, \boldsymbol{x}) = \sum_{t_2^{(2)}=1}^{T_2} \frac{1}{T_2} \boldsymbol{\lambda}(t_1, t^{(2)}, \boldsymbol{x})^\top \boldsymbol{y}^{(t_1, t^{(2)})}. \tag{3.18}$$

In this way, we can use the data $y^\star(\cdot)$ with $t^{(1)} = 1$ to estimate the average effect $\widehat{A}^{AK}(t^{(1)} = 1, \boldsymbol{x})$ and the new deviation $y_1^\star(\cdot)$ computed by subtracting $\widehat{A}^{AK}(t^{(1)} = 1, \boldsymbol{x})$ from $y^\star(\cdot)$. Similarly, we regard the deviation after taking out the effects of both $t^{(1)}$ and $t^{(2)}$ as $y_{1,2}^\star(\cdot)$. To estimate the interaction effect of $t^{(1)}$ and $t^{(2)}$, we use the data $y_{1,2}^\star(\cdot)$ with a fixed value of $(t^{(1)}, t^{(2)})$ to predict the response $y_{1,2}^\star(t^{(1)}, t^{(2)}, \cdot)$. For example, if the computer experiment has two qualitative inputs and we are interested in the prediction at $(t^{(1)}, t^{(2)}) = (1, 1)$, the prediction of the interaction effect will use the data $y_{1,2}^\star(\cdot)$ at $(t^{(1)}, t^{(2)}) = (1, 1)$. We regard the prediction for $y_{1,2}^\star(1, 1, \cdot)$ as the interaction effect at $(t^{(1)}, t^{(2)}) = (1, 1)$.

The prediction of $y(t_1, t_2, \boldsymbol{x}_0)$ (for $t_1 \in \{1, \ldots, T_1\}$, $t_2 \in \{1, \ldots, T_2\}$, and $\boldsymbol{x}_0 \in [0, 1]$) with the estimated main and interaction effects is computed as

$$\widehat{y}^{HAK}(t_1, t_2, \boldsymbol{x}_0) = \widehat{A}^{AK}(\boldsymbol{x}_0) + \widehat{A}^{AK}(t_1, \boldsymbol{x}_0) + \widehat{A}^{AK}(t_2, \boldsymbol{x}_0) + \widehat{y}_{1,2}^\star(t_1, t_2, \boldsymbol{x}_0). \tag{3.19}$$

It is possible that only the effect of one qualitative input is believed to be significant. Then one needs to estimate the overall average effect and the main effect of one qualitative input only. For example, if only the main effect of $t^{(1)}$ is believed to be significant, then the HAK prediction of $y(t^{(1)} = t_1, t^{(2)} = t_2, \boldsymbol{x}_0)$ is

$$\widehat{y}^{HAK}(t_1, t_2, \boldsymbol{x}_0) = \widehat{A}^{AK}(\boldsymbol{x}_0) + \widehat{A}^{AK}(t_1, \boldsymbol{x}_0) + \widehat{y}_1^{\star}(t_1, t_2, \boldsymbol{x}_0). \tag{3.20}$$

3.3.2 The HAK Predictor for Computer Experiments Having an Arbitrary Number of Qualitative Inputs

The idea of the HAK predictor for multivariate qualitative input factors is to construct the predictor using the estimated average effect and the estimated important main/interaction effects. The prediction is obtained by summing up the estimated effects and the prediction of the corresponding deviation. Before applying this method, one should determine which main/interaction effects are significant by using expert knowledge or by conducting sensitivity analysis (Saltelli, Chan and Scott (2000)).

Specifically, suppose the d dimensional quantitative input is $\boldsymbol{x} \in [0,1]^d$ and there are K quantitative inputs are $t^{(1)}, \ldots, t^{(K)}$, where the kth input $t^{(k)}$ has T_k levels for all $k = 1, \ldots, K$. We construct a new qualitative input with $\prod_{k=1}^{K} T_k$ levels corresponding to the $\prod_{k=1}^{K} T_k$ combinations of the K qualitative inputs. Then we compute $\widehat{A}^{AK}(\cdot)$ and the deviation $y^{\star}(\cdot)$ following Steps 1–6 in Section 3.2.3.

For each of the important main/interaction effects, our approach uses the data at certain level(s) of the qualitative input(s) to compute the effect and the corresponding new deviation. The HAK predictor is the sum of the average effects and the prediction of the corresponding deviation. For example, if J main and (or) interaction effects

are believed to be significant, the prediction of $y(t^{(1)} = t_1, \ldots, t^{(K)} = t_K, \boldsymbol{x}_0)$ is

$$\widehat{y}^{HAK}(t_{0,1}, \ldots, t_{0,K}, \boldsymbol{x}_0) = \widehat{A}^{AK}(\boldsymbol{x}_0) + \sum_{j=1}^{J} \widehat{A}_j(\boldsymbol{x}_0) + \widehat{y}^{\star}_{J+1}(t_{0,1}, \ldots, t_{0,K}, \boldsymbol{x}_0), \quad (3.21)$$

where $\widehat{A}_j(\boldsymbol{x}_0)$ denotes the jth estimated main or interaction effect and $\widehat{y}^{\star}_{J+1}(t_{0,1}, \ldots, t_{0,K}, \boldsymbol{x}_0)$ denotes the prediction of

$$y^{\star}_{J+1}(t_{0,1}, \ldots, t_{0,K}, \boldsymbol{x}_0) = y(t_{0,1}, \ldots, t_{0,K}, \boldsymbol{x}_0) - \left(\widehat{A}^{AK}(\cdot) + \sum_{j=1}^{J} \widehat{A}_j(\cdot) \right).$$

3.3.3 An Example Having One Quantitative Input and Two Qualitative Inputs

Suppose that both qualitative inputs have two levels and the quantitative input is in $[0, 1]$. We predict the observations at combination $(t^{(1)}, t^{(2)}) = (1, 1)$, where 4 data points were acquired. At each of the other three $(t^{(1)}, t^{(2)})$ combinations, 10 data points were acquired. We made the 10 input points and the 4 input points equally spaced over $[0, 1]$ by using LHDs. We generated data $y(t^{(1)}, t^{(2)}, \boldsymbol{x})$ using the following four equations:

$$y(1, 1, x) = 0.3x + 0.1 \sin(2.5\pi x) + 0.5(x - 0.5)^2; \quad (3.22)$$

$$y(2, 1, x) = 0.1 + 0.3x + 0.1 \sin(2.5\pi x) + 2.5(x - 0.5)^4 - 0.4x^5; \quad (3.23)$$

$$y(1, 2, x) = 0.2 + 0.3x + 0.1 \sin(2\pi x) + 0.5(x - 0.5)^2; \quad (3.24)$$

$$y(2, 2, x) = 0.3 + 0.3x + 0.1 \sin(2\pi x) + 2.5(x - 0.5)^4 - 0.4x^5. \quad (3.25)$$

Figure 3.3 shows the observations and the true curves. We see that the curve at $(t^{(1)}, t^{(2)}) = (1, 1)$ shares the sine curve trend $0.1 \sin(2.5\pi x)$ with the curve at $(t^{(1)}, t^{(2)}) = (2, 1)$; the $(t^{(1)}, t^{(2)}) = (1, 1)$ curve also shares the quadratic trend $(0.5(x - 0.5)^2)$ with the curve at $(t^{(1)}, t^{(2)}) = (1, 2)$. An ideal predictor should be able to capture these two trends.

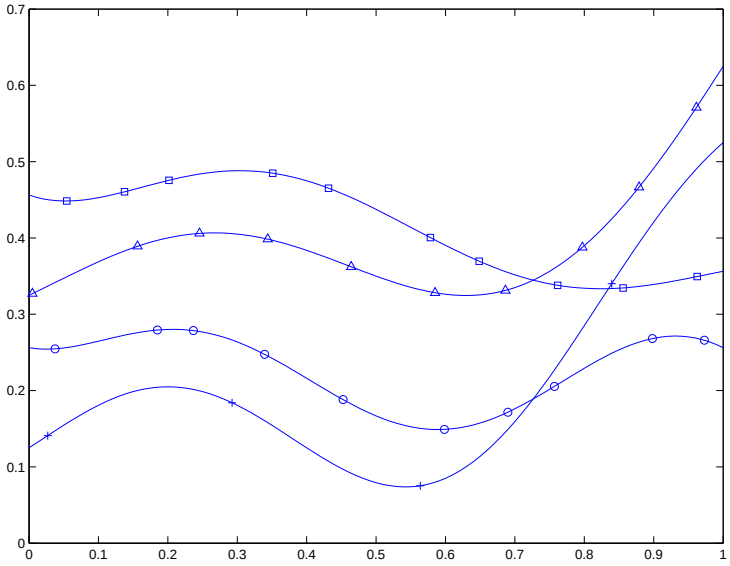

Figure 3.3: Raw data and the four true curves. Observations on the curves at level $(1,1)$, $(2,1)$, $(1,2)$, and $(2,2)$ are denoted by plus signs, circles, triangles, and squares, respectively.

We apply three predictors to this problem. The first predictor $\widehat{y}^{HQV}(\cdot)$ is the HQQV predictor described in Chapter 2 (after converting the two qualitative variables into a single one with 4 levels). The second is the HAK predictor $\widehat{y}_1^{HAK}(\cdot)$ that estimates $A^{AK}(\cdot)$ only. The third predictor is another HAK predictor $\widehat{y}_{1,2}^{HAK}(\cdot)$ that estimates $A^{AK}(\cdot)$ as well as the main and the interaction effects.

The RMSPEs of $\widehat{y}^{HQV}(\cdot)$, $\widehat{y}_1^{HAK}(\cdot)$, and $\widehat{y}_{1,2}^{HAK}(\cdot)$ at 101 input points $x \in \{0, 0.01, \ldots, 0.99, 1\}$ are 0.0384, 0.0237, and 0.0182. Thus, $\widehat{y}_1^{HAK}(\cdot)$ and $\widehat{y}_{1,2}^{HAK}(\cdot)$ have relative improvement rates of 38.2% ($\approx \frac{0.0384-0.0237}{0.0384}$) and 52.6% ($\approx \frac{0.0384-0.0182}{0.0384}$) over $\widehat{y}^{HQV}(\cdot)$. Figure 3.4 depicts the true curve at $(t^{(1)}, t^{(2)}) = (1,1)$ and the predicted $y(t^{(1)}, t^{(2)}, x)$ at $(t^{(1)}, t^{(2)}, x) = (1,1,0), (1,1,0.01), \ldots, (1,1,0.99), (1,1,1)$ using $\widehat{y}^{HQV}(\cdot)$, $\widehat{y}_1^{HAK}(\cdot)$,

and $\widehat{y}_{1,2}^{HAK}(\cdot)$. Visually we can see that $\widehat{y}_{1,2}^{HAK}(\cdot)$ is closer to the truth than $\widehat{y}^{HQV}(\cdot)$ and $\widehat{y}_{1}^{HAK}(\cdot)$.

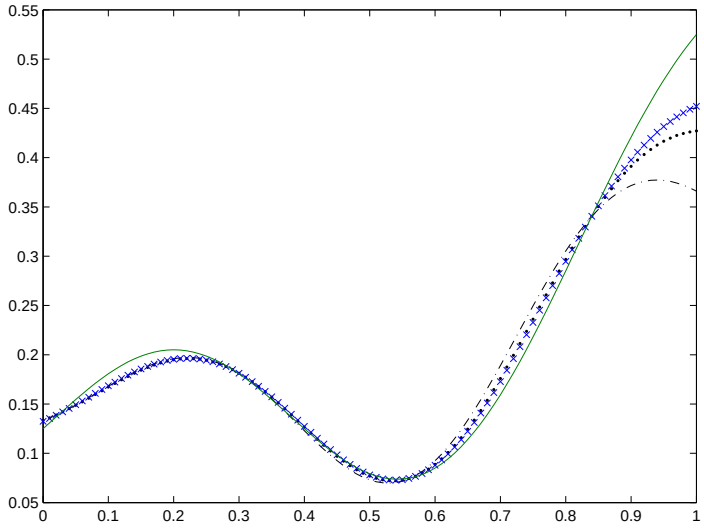

Figure 3.4: The true response $y(1,1,\cdot)$ (the solid curve), $\widehat{y}^{HQV}(1,1,\cdot)$ (the dashed dots), $\widehat{y}_{1}^{HAK}(1,1,\cdot)$ (the solid points), and $\widehat{y}_{1,2}^{HAK}(1,1,\cdot)$ (the x-mark).

In summary, $\widehat{y}_{1,2}^{HAK}(1,1,\cdot)$ has the smallest predictive error and both the HAK predictors have better predictive accuracies than the HQQV predictor having one qualitative input with four levels, which implies that the HAK predictor can effectively capture the common trend as well as the main/interaction effects.

3.4 An Application of the HAK Predictor to a Hip Resurfacing System

Long and Bartel (2006) described a hip resurfacing system whose shell geometry is representative of several designs of the prosthetic devices including the Brimingham$^{\text{TM}}$

(Smith and Nephew, London, England), the Conserve Plus (Wright Medical Technology, Arlington, TN), and the Durom^{TM} (Zimmer, Warsaw, IN). This implant was particularly designed for males under 60 because they were the main patients in the hip resurfacing surgery. One cadaveric bone was taken from a 37 year old male donor (Bone 1) and one from a 47 year old male donor (Bone 2). The two upper panels in Figure 3.5 (source: Long (2008)) show the left side of the two bones. The bones are of different sizes. The femur's head diameters are 50mm and 46mm. The neck lengths are 55mm and 54mm. The neck-shaft angles are 130^o and 132^o. The head-to-neck diameter ratio are 1.4 and 1.3. Given the differences, two finite element analysis computer codes were developed to simulate the two resurfacing proximal femurs. The two lower panels in Figure 3.5 (source: Long (2008)) show the simulated resurfacing proximal femurs drawn using a finite element mesh.

One of the goals of the computer simulation is to approximate the conditions causing damage (or abnormal behavior) to the resurfaced bone. The principal strain at the edge of the femoral resurfacing component was considered to be an important quantity determining the femur's function in that a minimum principal strain (MinPS) lower than -0.004 was believed to result in the resurfaced system's being under a high risk of malfunction (Long and Bartel (2006)). Thus a computer code was developed to simulate the MinPS. Among the inputs to the computer experiment, eight were considered significant so that we include these eight inputs in our analysis. Among them, five inputs were quantitative and three were qualitative with nominal values. Each of the three qualitative inputs has two levels. The five quantitative inputs were

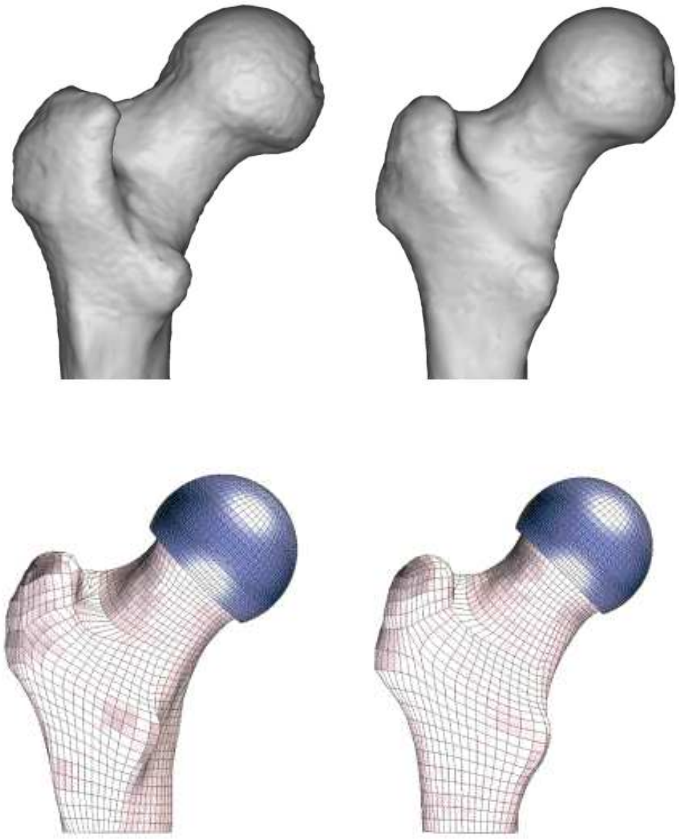

Figure 3.5: Two bones and the finite element models of the two bones with implants. Upper left panel: Bone 1; upper right panel: Bone 2; lower left panel: the finite element model of Bone 1 with its implant; lower right panel: the finite element model of Bone 2 with its implant. Source: Long (2008).

1. Density-modulus function weight (W): a parameter governing the choice of modulus-density relationship; W is assumed to have a symmetric triangular distribution having mean 0.5 and support $[0, 1]$;

2. In-plane head load angle (θ_i): a angle that describes the head load direction; θ_i distributed as truncated normal with mean 0, standard error 5.5, and the lower and upper bounds being -11 and 11;

3. Out-of-plane head load angle (θ_o): an angle describing the head load direction; θ_o was distributed as truncated normal with mean 0, standard error 1.0, and the lower and upper bounds being -2 and 2. (The variations in the head load direction was roughly planar so that a least-square plane was fit and the head load direction can be uniquely described by the in-plane and out-of-plane angles.)

4. Abductor-head load ratio (A): the abductor load at the point in time where the peak head load occurs; A is taken to be a linear function of θ_i plus a random noise distributed as truncated normal with mean 0, standard deviation $\sqrt{0.064 + 0.142}$, and the lower and upper bounds $-2 \times \sqrt{0.064 + 0.142}$ and $2 \times \sqrt{0.064 + 0.142}$, respectively.

5. Stem friction coefficient (μ): the Coulomb friction coefficient along the stem-bone interface where μ is a design input in $[0.1, 0.5]$.

The three qualitative inputs were

1. Bone (B): a qualitative input with value 0 corresponding to the 37 year old donor and value 1 corresponding to the 47 year old donor.

2. Stem-stem hole geometry (S): an input with value 0 corresponding to a design having a 0.5^o tapered stem in a 0.5^o tapered hold (line to line) and with value 1 corresponding to a design with a 0.5^o tapered stem in a straight hole (not line to line).

3. Shell fixation (F): an input with value 0 corresponding to a bonded shell surface having a displacement compatible interface and with value 1 corresponding to a deboned shell interface having a Coulomb friction interface.

The training data set in our analysis consisted of 80 points. The design matrix was obtained by the following three steps; i.e.,

1 Simulating 10 realizations of $(W, \theta_i, \theta_o, A)$ from their distributions;

2 Assigning each of the 8 combinations of B, S, and F to the 10 realizations;

3 Making the design of μ to be equally spaced in $[0.1, 0.5]$ for each 10 runs having a same (B, S, F) combination.

The test data set consisted of 10 inputs with $(B, S, F, \mu) = (0, 0, 0, 0.3)$ whose outputs were thought to be close to -0.004 based on a preliminary analysis. As described before, the predictive accuracy at these inputs were critical for the study of the resurfacing system. Thus, we attempt to develop a predictor with high predictive accuracy for the outputs less or equal to -0.004.

Using the sensitivity analysis, we detect that the significant effects are the main effects of B, S, and F, and the interaction effect between B and S. We implement and compare three HAK predictors by investigating their predictive accuracies for predicting the responses in the testing data set. The first predictor estimates only the common trend of the new qualitative input having 8 levels. The second predictor estimates the common trend and the main effects of B, S, and F. The third predictor estimates the common trend, main effects, as well as the interaction effect between B and S.

The cross validation procedure shows that none of the levels other than $(B, S, F) = (0, 0, 0)$ is selected when estimating the overall average effect, which indicates that the eight response surfaces, corresponding to the eight combinations of (B, S, F),

have no common trend. Thus, the common trend does not help improve the predictive accuracy of $\widehat{y}^{HAK}$ over $\widehat{y}^{HQV}$. Table 3.1 lists the true responses, the predictions, and the RMSPEs of $\widehat{y}^{HAK}_{1,2,3}$ and $\widehat{y}^{HAK}_{1,2,3,12}$. It shows that by combining the main and the interaction effects, the RMSPEs over the ten points have been decreased from 0.0005309 to 0.0003832 and to 0.0002594. The improvement rates of $\widehat{y}^{HAK}_{1,2,3}$ and $\widehat{y}^{HAK}_{1,2,3,12}$ over $\widehat{y}^{HAK}$ are $27.8\% = (0.0005309 - 0.0003832)/0.0005309$ and $51.1\% = (0.0005309 - 0.0002594)/0.0005309$. The third HAK predictor, which estimates the significant main and interaction effects, is therefore the most accurate predictor. Using $\widehat{y}^{HAK}_{1,2,3,12}$, one could accurately predict the minimum principal strain with a limited number of simulation runs and thus can reasonably design the implant to let MinPS be higher than -0.004 so that the resurfacing system would be more likely to work well.

Testing data cases	$y(\cdot)$	$\widehat{y}^{HAK}$ (or $\widehat{y}^{HQV}$)	$\widehat{y}^{HAK}_{1,2,3}$	$\widehat{y}^{HAK}_{1,2,3,12}$
1	-0.003850	-0.0035321	-0.0042721	-0.0040023
2	-0.003657	-0.0033484	-0.0042359	-0.0039446
3	-0.003946	-0.0034618	-0.0040383	-0.0038851
4	-0.003864	-0.0034843	-0.0043196	-0.0040240
5	-0.003766	-0.0030408	-0.0037451	-0.0036372
6	-0.003799	-0.0034083	-0.0040588	-0.0038231
7	-0.004363	-0.0034644	-0.0041865	-0.0039190
8	-0.003340	-0.0033852	-0.0040799	-0.0038706
9	-0.004165	-0.0034369	-0.0041192	-0.0039634
10	-0.003674	-0.0031988	-0.0039753	-0.0037082
RMSPE	—	**0.0005309**	**0.0003832**	**0.0002594**

Table 3.1: True responses of the testing data inputs and predictions of $\widehat{y}^{HAK}$, $\widehat{y}^{HAK}_{1,2,3}$, and $\widehat{y}^{HAK}_{1,2,3,12}$.

3.5 Summary and Future Research

This chapter proposes a methodology for predicting the output from computer experiments having both quantitative and qualitative inputs. The method can deal with an arbitrary number of qualitative inputs. The examples in Sections 3.2.5, 3.3.3, and 3.4 demonstrate that the HAK predictor captures the common shape of responses at different levels if such a common shape exists, and uses this to improve the overall prediction. In the proposed cross validation procedure, the HAK predictor uses only the data at the levels that share the similar trend of the data at the level for prediction.

For future research on the HQQV ANOVA kriging predictor, one issue is to construct a predictive (or credible) interval for the HAK predictor. Another is to extend this method to predict the response from a physical experiment using the data from both the physical experiment and one or more computer simulation codes having the same set of quantitative and qualitative inputs. To predict physical experiments, the following result can be used.

Proposition 3. Suppose that

1. There are $T - 1$ computer codes simulating the physical experiment;

2. The computer experiments are unknown deterministic functions and the physical experiment is a sum of an unknown deterministic function and a random error;

3. The input variables to the computer experiments and the physical experiment are identical and the inputs are all quantitative.

If we create a discrete variable $t \in \{1, \ldots, T\}$ and regard the physical experiment as being at level t_0 and the computer experiments as being at the other $T-1$ levels, then **Proposition 1** in Section 3.2.2 holds; i.e.,

$$V^{OK}(t_0, \boldsymbol{x}_0) - V^{AE}(t_0, \boldsymbol{x}_0) = \frac{2}{T}BD - \frac{1}{T^2}D^2,$$

where

$$B = \boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \boldsymbol{y}^{(t_0)} - y(t_0, \boldsymbol{x}_0)$$

and

$$D = \sum_{t \neq t_0} \left[\boldsymbol{\lambda}(t_0, \boldsymbol{x}_0)^\top \begin{pmatrix} y(t, \boldsymbol{x}_1^{(t_0)}) \\ \vdots \\ y(t, \boldsymbol{x}_{n_{t_0}}^{(t_0)}) \end{pmatrix} - y(t, \boldsymbol{x}_0) \right].$$

Proof. Suppose that in the physical experiment, the noise term $\epsilon(\boldsymbol{x})$ has mean 0 and variance σ^2. The predictive error of $\widehat{y}^{OK}(t_0, \boldsymbol{x}_0)$ is

$$V^{OK}(t_0, \boldsymbol{x}_0) = E(\widehat{y}^{OK}(t_0, \boldsymbol{x}_0) - y(t_0, \boldsymbol{x}_0))^2 = B^2 + \sigma^2. \tag{3.26}$$

By (3.11), $\widehat{y}^{AE}(t_0, \boldsymbol{x}_0) = \widehat{y}^{OK}(t_0, \boldsymbol{x}_0) - \frac{1}{T}D$ so that the predictive error of $\widehat{y}^{AE}(t_0, \boldsymbol{x}_0)$ is

$$V^{AE}(t_0, \boldsymbol{x}_0) = E(\widehat{y}^{AE}(t_0, \boldsymbol{x}_0) - y(t_0, \boldsymbol{x}_0))^2 = (B - D)^2 + \sigma^2. \tag{3.27}$$

By (3.26) and (3.27),

$$V^{OK}(t_0, \boldsymbol{x}_0) - V^{AE}(t_0, \boldsymbol{x}_0) = B^2 - (B - \frac{1}{T}D)^2 = \frac{2}{T}BD - \frac{1}{T^2}D^2. \quad \blacksquare$$

Research on combining Proposition 3 with a plausible statistical model and with a procedure that selects levels to be used to construct the HAK predictor for the physical experiment is underway.

CHAPTER 4

SIMULTANEOUS CALIBRATION AND TUNING FOR COMPUTER EXPERIMENTS

4.1 Introduction

Many computer codes have been developed and used in settings where physical experiments are also available for describing the true input/output relationship of interest (chapter 1 of Santner et al. (2003) and chapter 1 of Fang et al. (2005)). In this chapter, we are interested in settings where both types of data are available but the computer code takes sufficiently long to run so that the number of computer code runs is limited.

In this setting, statistical models are needed to predict the output from the computer code and the physical experiment. Such statistical models must incorporate different types of input variables required by computer codes, such as control inputs, tuning parameters, and calibration parameters. Next we review current methods for tuning and calibration and discusses their limitations.

Park (1991) and Cox et al. (1996) set tuning parameters to make the computer code output fit the physical observations as closely as possible in an integrated prediction error sense. Approaches that search for "best" tuning parameters gives the

selection of the tuning parameters without providing uncertainties associated with the proposed ideal values of the tuning parameters.

Previous proposals for calibration have been primarily Bayesian. Their goal has been to simulate the posterior distributions of the calibration parameters and to predict unknown responses. Craig, Goldstein, Seheult and Smith (1996) and Craig, Goldstein, Rougier and Seheult (2001) developed *a Bayesian linear forecasting method*, which can be seen as an approximation to a fully Bayesian analysis. Kennedy and O'Hagan (2001) described a Bayesian calibration framework based on a model having a bias function and a (modular) Bayesian analysis. Higdon et al. (2004) proposed a fully Bayesian implementation for the model in Kennedy and O'Hagan (2001). Gattiker, Williams and Rightley (2005) developed a methodology for calibration when the outputs are multivariate. Gattiker (2005) implemented the model in Higdon et al. (2004) and Gattiker et al. (2005) in MATLAB. Loeppky, Bingham and Welch (2006) proved that whose procedure would lead to asymptotically correct estimation of the calibration parameters if the true values of the calibration parameters are such that they reduce the bias of the computer simulation to zero. We will discuss a lemma in Loeppky et al. (2006) in Section 4.4.1.

We note that some models used to "validate" computer experiments (or "assess" the usefulness of a computer code) are similar to the models for Bayesian calibration. For example, Bayarri, Berger, Paulo, Sacks, Cafeo, Cavendish, Lin and Tu (2007b) presented a framework with a Bayesian model for validation and Bayarri, Berger, Cafeo, Garcia-Donato, Liu, Palomo, Parthasarathy, Paulo, Sacks and Walsh (2007a) used wavelet decomposition based on the same Bayesian model to validate computer experiments having multivariate or functional outputs.

Because the methods above focus on either tuning or calibration, these methods may suffer limitations in applications where there are both tuning and calibration parameters. Specifically, using either methodology in a problem involving both types of parameters can be problematic. For example, if one applies a method for tuning to a problem involving both tuning and calibration parameters, it becomes unable to quantify the possible uncertainties of the calibration parameters. If one uses a calibration methodology regarding both tuning and calibration parameters as calibration parameters, one can get biased estimations and/or undesired estimated uncertainties of the tuning and calibration parameters as we will show below. These misleading estimations and estimated uncertainties can lead to large prediction errors.

To demonstrate the limitations described above, we will use the Bayesian calibration program (`gpmsa`) described in Gattiker (2005) to set tuning and calibration parameters for an application described in Rawlinson et al. (2006). This project originally compared the damage in two knee implants (the Install-Burstein (IB) manufactured by Zimmer, Inc. and the Optetrak produced by Exactech, Inc.). The responses from the physical experiment for this study were the measurements made on the kinematics and kinetics of knees tested in an Instron-Stanmore KC1 testing device, a "knee simulator." The loading (magnitude, angle, and rate) and knee design were modeled in a finite element analysis (FEA) computer code whose output included the kinematics, kinetics, and stresses experienced by the knee component. Specifically, the anterior-posterior displacement (APD), an important kinematic output of fore-aft motion during knee function, was measured in both the knee simulator and the computer simulation. The APD was roughly proportional to the anterior-posterior force that, coupled with the vertical joint load, contributed to damage of the

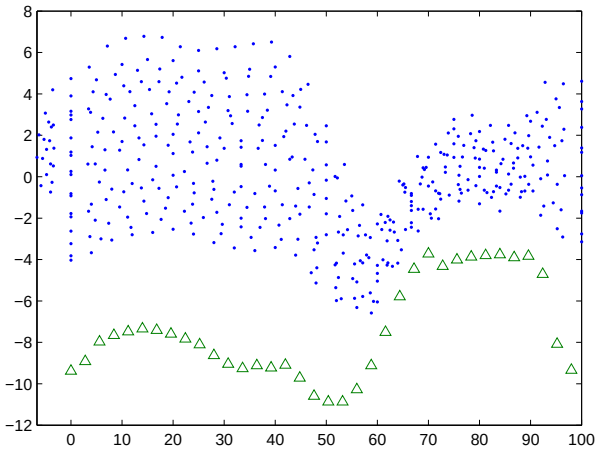

Figure 4.1: Scatter plot of the measured APD over the gait cycle, from the knee simulator (triangles) and the FEA computer code (dots).

prosthesis. This computer experiment had one control variable (the percentile in gait cycle), two tuning parameters (finite element mesh density and load discretization), and two calibration parameters (friction and initial position).

In our application we consider only the IB knee implant. The design of the computer and the physical experiments were roughly Maximin Latin Hypercube designs (McKay et al. (1979)). A pilot study of the APD for this design acquired 439 observations from the computer code and 36 observations from the physical experiment for the purpose of tuning and calibration. Figure 4.1 depicts the training data. A previous sensitivity analysis (Saltelli et al. (2000)) found that APD was not sensitive to friction and mesh density. Therefore below we only study load discretization (a tuning parameter) and initial position (a calibration parameter).

Using `gpmsa`, we regarded both load discretization and initial position as calibration parameters, whose prior distributions were near uniform distribution. We ran

`gpmsa` with 8000 burn-in iterations and 2000 production runs, and based our inferences on 100 equally-spaced values of each parameter taken from the production runs. Figure 4.2 shows the simulated posterior distributions of these two parameters. The posterior distribution of load discretization is bimodal and the one of initial position has the mode on lower values but has large variation. Thus, plausible values of the tuning and calibration parameters remain unclear, which makes the determination of their values impossible. We will demonstrate that our methodology can significantly

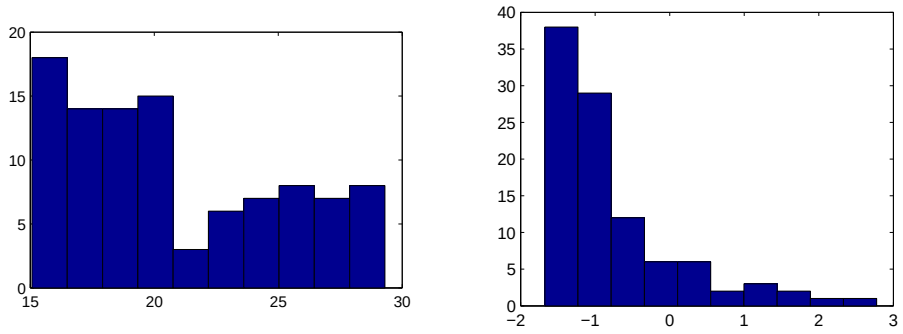

Figure 4.2: Simulated posterior distributions of load discretization (the left panel) and initial position (the right panel).

improve the selection of the tuning and calibration parameters in this application.

Chapter 4 is organized as follows: Section 4.2 introduces a hierarchical Bayesian model describing the response from a physical experiment together with the output from the computer simulation code having both tuning and calibration parameters. Section 4.3 proposes our methodology for setting these parameters simultaneously. Section 4.4 compares our methodology with an approach that treats all parameters as calibration parameters. Section 4.5 summarizes this chapter.

4.2 A Hierarchical Bayesian Model for Tuning and Calibration

Let $\boldsymbol{x}$ denote the vector of control variables, $\boldsymbol{t}$ denote the vector of tuning parameters, and $\boldsymbol{t}^\star$ denote the "best" values of the tuning parameters, where $\boldsymbol{t}^\star$ will be defined in Section 4.3. (Warning: in Chapters 2 and 3, $\boldsymbol{t}$ denotes qualitative inputs, while in this chapter $\boldsymbol{t}$ denotes tuning parameters.) Let $\boldsymbol{c}$ denote the vector of calibration parameters and $\boldsymbol{\theta}_c$ denote the *unknown* true values of the calibration parameters. Assume $\boldsymbol{x} \in [0,1]^{p_x}$, $\boldsymbol{t} \in [0,1]^{p_t}$, and $\boldsymbol{c} \in [0,1]^{p_c}$ or they can be so scaled.

Let $\{(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i),\ y^s(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i);\ i = 1, 2, \ldots, n_s\}$ denote the training data from the computer experiment and $\{\boldsymbol{x}_j^p, y^p(\boldsymbol{x}_j^p);\ j = 1, 2, \ldots, n_p\}$ denote the training data from the physical experiment. Here n_s and n_p are the numbers of runs for each type of data. A more complete notation for the physical experiment response is $y^p(\boldsymbol{x}, \boldsymbol{\theta}_c)$; throughout this paper, $y^p(\boldsymbol{x})$ and $y^p(\boldsymbol{x}, \boldsymbol{\theta}_c)$ are equivalent. Let $\boldsymbol{y}^s$ be the $n_s \times 1$ vector with ith element $y^s(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)$ and $\boldsymbol{y}^p$ be the $n_p \times 1$ vector with jth element $y^p(\boldsymbol{x}_j^p)$.

The model and the prior distribution proposed here are similar in spirit to the ones in Kennedy and O'Hagan (2001) and Higdon et al. (2005) except that our model incorporates both the tuning and calibration parameters. Roughly, the response from the physical experiment is the sum of the "true response" and a random noise; the output from the computer experiment is the difference between the true response and the code bias. The proposed model assumes that the output from the computer experiment and the bias can be described as draws from Gaussian stochastic processes. Specifically, we regard $y^p(\cdot)$ as a realization of

$$Y^p(\boldsymbol{x}) = \eta(\boldsymbol{x}, \boldsymbol{\theta}_c) + \epsilon(\boldsymbol{x}), \tag{4.1}$$

where $\eta(\boldsymbol{x}, \boldsymbol{\theta}_c) = E(Y^p(\boldsymbol{x}))$ is the true response at $\boldsymbol{x}$ and $\epsilon(\cdot)$ is the white noise Gaussian process with the mean 0 and the covariance $Cov(\epsilon(\boldsymbol{x}_1), \epsilon(\boldsymbol{x}_2)) = 0$ if $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ and $Cov(\epsilon(\boldsymbol{x}_1), \epsilon(\boldsymbol{x}_2)) = \sigma_\epsilon^2$ if $\boldsymbol{x}_1 = \boldsymbol{x}_2$. Thus $\boldsymbol{y}^p$ can be viewed as a realization of the random vector $\boldsymbol{Y}^p$ whose jth element is $Y^p(\boldsymbol{x}_j^p)$.

The proposed hierarchical Bayesian model views $y^s(\cdot)$ as a realization of the Gaussian stochastic process

$$Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = \boldsymbol{f}_Z^\top(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\boldsymbol{\beta}_Z + Z(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}). \tag{4.2}$$

The mean of $Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ is $\boldsymbol{f}_Z^\top(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\boldsymbol{\beta}_Z$, where $\boldsymbol{f}_Z^\top(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ is a vector of *known* regression coefficients and $\boldsymbol{\beta}_Z$ is a vector of *unknown* regression parameters. If $Y^s(\cdot)$ is assumed to have a constant mean, then $\boldsymbol{f}_Z^\top(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\boldsymbol{\beta}_Z = \beta_{Z,0}$. In (4.2), $Z(\cdot)$ is the stationary Gaussian stochastic process with the mean 0, variance σ_Z^2, and *product Gaussian correlation*

$$Cor(Z(\boldsymbol{x}_1, \boldsymbol{c}_1, \boldsymbol{t}_1), Z(\boldsymbol{x}_2, \boldsymbol{c}_2, \boldsymbol{t}_2)|\boldsymbol{\rho}_Z) = \prod_{i=1}^{p_x} \rho_{Z,x,i}^{4\left(x_i^1 - x_i^2\right)^2} \times \prod_{j=1}^{p_c} \rho_{Z,c,j}^{4\left(c_j^1 - c_j^2\right)^2} \times \prod_{k=1}^{p_t} \rho_{Z,t,k}^{4\left(t_k^1 - t_k^2\right)^2} \tag{4.3}$$

with correlation parameters $\boldsymbol{\rho}_Z = (\rho_{Z,x,1}, \ldots, \rho_{Z,x,p_x}, \rho_{Z,c,1}, \ldots, \rho_{Z,c,p_c}, \rho_{Z,t,1}, \ldots, \rho_{Z,t,p_t})^\top$. Thus $\boldsymbol{y}^s$ can be viewed as a realization of the random vector $\boldsymbol{Y}^s$ whose ith element is $Y^s(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)$.

We define $\delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ to be the bias of the simulation at $(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$; i.e.,

$$\delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = \eta(\boldsymbol{x}, \boldsymbol{\theta}_c) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}). \tag{4.4}$$

The model views $\delta(\cdot)$ as a draw from the Gaussian stochastic process

$$\Delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = \boldsymbol{f}_D^\top(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\boldsymbol{\beta}_D + D(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}), \tag{4.5}$$

where $\boldsymbol{f}_D(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ is a vector of known regression coefficients and $\boldsymbol{\beta}_D$ is a vector of unknown regression parameters. When $\Delta(\cdot)$ is assumed to have a constant mean,

$\boldsymbol{f}_Z^\top(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\, \boldsymbol{\beta}_Z = \beta_{D,0}$. Similar to $Z(\cdot)$, $D(\cdot)$ is assumed to be the stationary Gaussian stochastic process with the mean 0, variance σ_D^2, and correlation function

$$Cor(D(\boldsymbol{x}_1, \boldsymbol{c}_1, \boldsymbol{t}_1), D(\boldsymbol{x}_2, \boldsymbol{c}_2, \boldsymbol{t}_2)|\boldsymbol{\rho}_D) = \prod_{i=1}^{p_x} \rho_{D,x,i}^{4\left(x_i^1 - x_i^2\right)^2} \times \prod_{j=1}^{p_c} \rho_{D,c,j}^{4\left(c_j^1 - c_j^2\right)^2} \times \prod_{k=1}^{p_t} \rho_{D,t,k}^{4\left(t_k^1 - t_k^2\right)^2}, \tag{4.6}$$

where $\boldsymbol{\rho}_D = (\rho_{D,x,1}, \ldots, \rho_{D,x,p_x}, \rho_{D,c,1}, \ldots, \rho_{D,c,p_c}, \rho_{D,t,1}, \ldots, \rho_{D,t,p_t})^\top$. Finally, $Y^s(\cdot)$, $\Delta(\cdot)$, and $\epsilon(\cdot)$ are assumed to be mutually *independent.*

The parameters in our model are $\boldsymbol{\theta}_c$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_Z^\top, \boldsymbol{\beta}_D^\top)^\top$, $\boldsymbol{\sigma} = (\sigma_Z^2, \sigma_D^2, \sigma_\epsilon^2)^\top$, and $\boldsymbol{\rho} = (\boldsymbol{\rho}_Z^\top, \boldsymbol{\rho}_D^\top)^\top$. For a generic parameter (vector) $\boldsymbol{\nu}$, let $[\boldsymbol{\nu}]$ denote its prior distribution. We assume that the priors of the calibration parameters, process means, process variances, and process correlations are independent so that

$$[\boldsymbol{\theta}_c, \boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho}] = [\boldsymbol{\theta}_c] \times [\boldsymbol{\beta}] \times [\boldsymbol{\sigma}] \times [\boldsymbol{\rho}]. \tag{4.7}$$

In the examples below, vague priors are constructed for $\boldsymbol{\theta}_c$, $\boldsymbol{\beta}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\rho}$; of course, if expert knowledge is available it should be used to provide informative priors. In particular, because the calibration parameters are scaled to $[0,1]^{p_c}$, all the elements of $\boldsymbol{\theta}_c$ are taken to have an independent and identical truncated normal distribution with the mean 0.5, standard deviation 2, and support $[0,1]$. This prior is close to a uniform distribution on $[0,1]^{p_c}$. We take $\boldsymbol{f}_Z^\top(\cdot)\boldsymbol{\beta}_Z = \beta_{Z,0}$ and $\boldsymbol{f}_D^\top(\cdot)\boldsymbol{\beta}_D = \beta_{D,0}$ and set $\beta_{Z,0}$ to be the average of the outputs from the computer experiment and $\beta_{Z,0} + \beta_{D,0}$ to the average of the responses from the physical experiment, which are comparable to the degenerate priors for the process means assumed by Gattiker (2005). The prior distributions of the variance parameters are set to be independent inverse Gamma distributions. The notation $\mathrm{IG}(\alpha, \gamma)$ denotes the inverse gamma distribution with mean and the variance $1/[\gamma(\alpha-1)]$ and $1/[\gamma^2(\alpha-1)^2(\alpha-2)]$, respectively. We take

$\sigma_Z^2 \sim \text{IG}(\alpha_Z, \gamma_Z)$, $\sigma_\epsilon^2 \sim \text{IG}(\alpha_\epsilon, \gamma_\epsilon)$, and $\sigma_D^2 \sim \text{IG}(\alpha_D, \gamma_D)$ where the parameters are set data-adaptively.

In detail, the prior parameters of the IG priors are constructed by first computing the sample variances of the output from the computer experiment and the physical experiment; we denote these quantities by $\widehat{\sigma}_s^2$ and $\widehat{\sigma}_p^2$, respectively. Let σ_Z^2 have a mildly informative prior distribution by setting $(\alpha_Z, \gamma_Z) = (10, 0.1/\widehat{\sigma}_s^2)$ and σ_ϵ^2 have a prior with small magnitude and variation by taking $(\alpha_\epsilon, \gamma_\epsilon) = (1, 100/\widehat{\sigma}_s^2)$. We use the value of $(\widehat{\sigma}_p^2 - \widehat{\sigma}_s^2)$ to set the prior for σ_D^2. If $\widehat{\sigma}_p^2 > \widehat{\sigma}_s^2$, σ_D^2 is set to have a mildly informative prior by taking $(\alpha_D, \gamma_D) = (10, 0.1/(\widehat{\sigma}_p^2 - \widehat{\sigma}_s^2))$ while if $\widehat{\sigma}_p^2 \leq \widehat{\sigma}_s^2$, σ_D^2 is set to have a prior with small magnitude and variation where $(\alpha_D, \gamma_D) = (1, 100/\widehat{\sigma}_s^2)$. Finally, we take the prior for the correlation parameters to be independently and identically distributed with Beta$(1, 0.5)$ distribution which says that they have prior mean equal to $2/3$, prior model close to 1, and a rather diffuse support over $(0, 1)$.

We let $\boldsymbol{\phi} = (\boldsymbol{\beta}, \boldsymbol{\sigma}, \boldsymbol{\rho})$. Then the joint prior distribution of all the parameters is $[\boldsymbol{\theta}_c, \boldsymbol{\phi}]$. For a fixed $\boldsymbol{t}$, $Y^p(\boldsymbol{x}) = \eta(\boldsymbol{x}, \boldsymbol{\theta}_c) + \epsilon(\boldsymbol{x}) = y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) + \delta(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) + \epsilon(\boldsymbol{x})$. Thus the likelihood can be regarded as a function of $\boldsymbol{t}$ (and $(\boldsymbol{\theta}_c, \boldsymbol{\phi})$). The posterior density has the form

$$\begin{aligned} [\boldsymbol{\theta}_c, \boldsymbol{\phi} | \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}] &\propto [\boldsymbol{y}^p, \boldsymbol{y}^s | \boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}] \times [\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}] \\ &\propto [\boldsymbol{y}^p, \boldsymbol{y}^s | \boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}] \times [\boldsymbol{\theta}_c, \boldsymbol{\phi}]. \end{aligned}$$

The posterior density is proportional to the product of the joint prior density $[\boldsymbol{\theta}_c, \boldsymbol{\phi} | \boldsymbol{t}]$ and the likelihood $[\boldsymbol{y}^s, \boldsymbol{y}^p | \boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}]$. We can derive $[\boldsymbol{\theta}_c, \boldsymbol{\phi} | \boldsymbol{t}]$ by computing the prior densities of $[\boldsymbol{\theta}_c]$, $[\boldsymbol{\beta}]$, $[\boldsymbol{\sigma}]$, and $[\boldsymbol{\rho}]$ and applying (4.7). The density $[\boldsymbol{y}^s, \boldsymbol{y}^p |\, \boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}]$ is derivable analytically because $\boldsymbol{y}^s$ and $\boldsymbol{y}^p$ are viewed as observations from Gaussian

stochastic processes. Specifically, our model is

$$Y^s(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i) = \boldsymbol{f}_Z^\top(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)\boldsymbol{\beta}_Z + Z(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)$$

and

$$Y^p(\boldsymbol{x}_j^p) = Y^s(\boldsymbol{x}_j^p, \boldsymbol{\theta}_c, \boldsymbol{t}) + \Delta(\boldsymbol{x}_j^p, \boldsymbol{\theta}_c, \boldsymbol{t}) + \epsilon(\boldsymbol{x}_j^p)$$

for all $i = 1, \ldots, n_s$ and $j = 1, \ldots, n_p$. The mean values of $Y^p(\boldsymbol{x}_j^p)$ and $Y^s(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)$ are $\boldsymbol{f}_Z^\top(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)\boldsymbol{\beta}_Z + \boldsymbol{f}_D^\top(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i)\boldsymbol{\beta}_D$ and $\boldsymbol{f}_Z^\top(\boldsymbol{x}_j^p, \boldsymbol{\theta}_c, \boldsymbol{t})\boldsymbol{\beta}_Z$ respectively. The covariance matrix between $\boldsymbol{Y}^s$ and $\boldsymbol{Y}^p$ given $(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t})$ is an $n_s \times n_p$ matrix whose (i, j)th element is

$$Cov(Y^s(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i), Y^p(\boldsymbol{x}_j^p)|(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t})) = Cov(Z(\boldsymbol{x}_i^s, \boldsymbol{c}_i, \boldsymbol{t}_i), Z(\boldsymbol{x}_j^p, \boldsymbol{\theta}_c, \boldsymbol{t})|(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t})),$$

which can be computed using (4.3).

With the joint prior density and the likelihood, we simulate draws from $[\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]$ by implementing a Metropolis-Hastings (MH) algorithm, which updates each of the parameters with the specifications listed in Table 4.1. For all the parameters, the proposal distributions are uniform distributions. The centers of the uniform distributions are set to be the previous draws and the lengths of the uniform distributions are specified in the last column of Table 4.1. For any correlation parameter ρ, the proposal draws are made on $\xi = -4log(\rho)$. With the model and the simulated joint posterior distribution $[\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]$ for any given $\boldsymbol{t}$, the methodology for simultaneous tuning and calibration is proposed next.

Model parameter	Prior distribution	Support	Initial value	Length
All elements of $\boldsymbol{\theta}_c$	$\text{TN}(0.5, 2^2)$	$[0, 1]$	0.5	0.1
All elements of $\boldsymbol{\rho}_z$, $\boldsymbol{\rho}_\delta$	$\text{Beta}(1, 0.5)$	$(0, 1)$	$2/3$	0.3
σ_z^2	$\text{IG}(10, \frac{0.1}{\widehat{\sigma}_s^2})$	$(0, +\infty)$	$\frac{1}{\widehat{\sigma}_s^2}$	$\frac{0.2}{\widehat{\sigma}_s^2}$
σ_D^2, if $\widehat{\sigma}_s^2 > \widehat{\sigma}_p^2$	$\text{IG}(1, \frac{100}{\widehat{\sigma}_s^2})$	$(0, +\infty)$	$\frac{100}{\widehat{\sigma}_s^2}$	$\frac{10}{\widehat{\sigma}_s^2}$
σ_D^2, if $\widehat{\sigma}_s^2 < \widehat{\sigma}_p^2$	$\text{IG}(10, \frac{0.1}{\widehat{\sigma}_p^2 - \widehat{\sigma}_s^2})$	$(0, +\infty)$	$\frac{1}{\widehat{\sigma}_p^2 - \widehat{\sigma}_s^2}$	$\frac{0.1}{\widehat{\sigma}_p^2 - \widehat{\sigma}_s^2}$
σ_ϵ^2	$\text{IG}(1, \frac{100}{\widehat{\sigma}_s^2}$	$(0, +\infty)$	$\frac{100}{\widehat{\sigma}_s^2}$	$\frac{2}{\widehat{\sigma}_s^2}$

Table 4.1: Specifications of the Metropolis-Hastings algorithm. The four columns, from left to right, correspond to the prior distributions, the lower and upper bounds of the parameters as the program iterates, the initial values of the parameters, and the lengths of the uniform distributions. We let $\text{TN}(\mu, \sigma^2)$ on $[a, b]$ denote the truncated normal distribution with mean μ and variance σ^2 on the support $[a, b]$.

4.3 Methodology for Simultaneous Tuning and Calibration

4.3.1 The Discrepancy Function

First we select a discrepancy function between the computer code output and the physical experiment response. The discrepancy, regarded as a function of $(\boldsymbol{c}, \boldsymbol{t})$, is used to define the "true" tuning parameter. Three, of many, possible discrepancy functions are L_2 discrepancy $\int_0^1 \left(\eta(\boldsymbol{x}, \boldsymbol{\theta}_c) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\right)^2 d\boldsymbol{x}$, L_1 discrepancy $\int_0^1 |\eta(\boldsymbol{x}, \boldsymbol{\theta}_c) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})| d\boldsymbol{x}$, and L_∞ discrepancy $\max_{\boldsymbol{x}} |\eta(\boldsymbol{x}, \boldsymbol{\theta}_c) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})|$. In this article, we use L_2 discrepancy but our methodology can be applied with any other discrepancy function.

We let $S^2(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = (\eta(\boldsymbol{x}, \boldsymbol{\theta}_c) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}))^2$, then

$$S^2(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = \left[\delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) + (y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}))\right]^2 \tag{4.8}$$

by applying (4.4). If $\boldsymbol{\theta}_c = \boldsymbol{c}$ in (4.8), then $S^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) = \delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$ (because $y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) - y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) = 0$ with $\boldsymbol{\theta}_c = \boldsymbol{c}$) and so the L_2 discrepancy for $\boldsymbol{t}$ is $\int_0^1 S^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) d\boldsymbol{x} = \int_0^1 \delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) d\boldsymbol{x}$. Similarly, if $\boldsymbol{\theta}_c$ has a posterior distribution

$[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]$, the L_2 discrepancy for $\boldsymbol{t}$ is
$\int_0^1 \int_0^1 S^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]d\boldsymbol{\theta}_c d\boldsymbol{x} = \int_0^1 \int_0^1 \delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]d\boldsymbol{\theta}_c d\boldsymbol{x}$. We take the objective of tuning to be the estimation of

$$\boldsymbol{t}^\star = \underset{\boldsymbol{t}}{\operatorname{argmin}} \int_0^1 \int_0^1 \delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]d\boldsymbol{\theta}_c d\boldsymbol{x}. \tag{4.9}$$

The objective of calibration is to compute the posterior distribution $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}^\star]$; notice that posterior distribution used in calibration is at $\boldsymbol{t}^\star$, the true value of $\boldsymbol{t}$.

4.3.2 Simultaneous Tuning and Calibration

To review, the main idea is to estimate $\boldsymbol{t}^\star$ by a proposed $\boldsymbol{t}$ that minimizes the estimated squared discrepancy function and to conduct calibration with the estimated $\boldsymbol{t}^\star$. Next, we describe our estimation of $\delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$, $\boldsymbol{t}^\star$, and $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}^\star]$ and then summarize the methodology as a 3-step procedure.

First, we estimate $\delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$ by the posterior mean $E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})$, which is the Bayes predictor of $\delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$ under *squared error loss* (Santner et al. (2003)). Notice that $E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})$ is equal to

$$[E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})]^2 + Var(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}); \tag{4.10}$$

both $[E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})]^2$ and $Var(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})$ can be derived analytically based on the probability density of the multivariate normal distribution.

We provide the derivation next. Following (4.5) and (4.6), we compute

$$[E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})]^2 = [\boldsymbol{f}_D^\top(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})\boldsymbol{\beta}_D + \boldsymbol{\Sigma}_{DZ}^\top \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{D}_{\boldsymbol{t}}]^2$$

and

$$Var(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}) = \sigma_D^2 - \boldsymbol{\Sigma}_{DZ}^\top \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{\Sigma}_{DZ},$$

where $\boldsymbol{\Sigma}_{DZ}$ is the vector of covariances between $\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$ and $(\boldsymbol{Y}^{s\top}, \boldsymbol{Y}^{p\top})^\top$, $\boldsymbol{\Sigma}_{ZZ}$ is the variance-covariance matrix of $(\boldsymbol{Y}^{s\top}, \boldsymbol{Y}^{p\top})^\top$, and $\boldsymbol{D_t}$ is the difference between the data and the expectation; i.e., $\boldsymbol{D_t} = (\boldsymbol{y}^{p\top}\boldsymbol{y}^{s\top})^\top - (E(\boldsymbol{Y}^p|\boldsymbol{t})^\top E(\boldsymbol{Y}^s|\boldsymbol{t})^\top)^\top$. We list the elements in $\boldsymbol{\Sigma}_{\Delta Z}$, $\boldsymbol{\Sigma}_{ZZ}$, and $\boldsymbol{D_t}$ below.

- The ith element of $\boldsymbol{\Sigma}_{DZ}$ is $Cov(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}),\ Y^p(\boldsymbol{x}_i^p)\ |(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t})) = Cov(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}),\ \Delta(\boldsymbol{x}_i^p, \boldsymbol{\theta}_c,\ \boldsymbol{t})|(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}))$ for $i = 1, 2, \ldots, n_p$. The ith element of $\boldsymbol{\Sigma}_{DZ}$ is $Cov(\Delta(\boldsymbol{x},\ \boldsymbol{\theta}_c, \boldsymbol{t}), Y^s(\boldsymbol{x}_{i-n_p}^s,\ \boldsymbol{c}_{i-n_p},\ \boldsymbol{t}_{i-n_p})|\ (\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t})) = 0$ for $i = n_p + 1, n_p + 2, \ldots, n_p + n_s$.

- We parametrize $\boldsymbol{\Sigma}_{ZZ}$ as

$$\boldsymbol{\Sigma}_{ZZ} = \begin{pmatrix} \boldsymbol{\Sigma}_{PP} & \boldsymbol{\Sigma}_{PS} \\ \boldsymbol{\Sigma}_{PS}^\top & \boldsymbol{\Sigma}_{SS} \end{pmatrix}, \text{ where} \tag{4.11}$$

  - $\boldsymbol{\Sigma}_{PP}$ is an $n_p \times n_p$ matrix whose (i_1, i_2)th entry is $Cov(Y^p(\boldsymbol{x}_{i_1}^p), Y^p(\boldsymbol{x}_{i_2}^p)|\ (\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}))$;
  - $\boldsymbol{\Sigma}_{PS}$ is an $n_p \times n_s$ matrix whose (i, j)th entry is $Cov(Y^p(\boldsymbol{x}_i^p),\ Y^s(\boldsymbol{x}_j^s,\ \boldsymbol{c}_j,\ \boldsymbol{t}_j)|(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}))$;
  - $\boldsymbol{\Sigma}_{SS}$ is an $n_s \times n_s$ matrix whose (j_1, j_2)th entry is $Cov(Y^s(\boldsymbol{x}_{j_1}^s,\ \boldsymbol{c}_{j_1},\ \boldsymbol{t}_{j_1}),\ Y^s(\boldsymbol{x}_{j_2}^s,\ \boldsymbol{c}_{j_2},\ \boldsymbol{t}_{j_2})\ |(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}))$, for all $i, i_1, i_2 = 1, 2, \ldots, n_s$ and $j, j_1, j_2 = 1, 2, \ldots, n_p$.

- The ith element of $\boldsymbol{D_t}$ is $y^p(\boldsymbol{x}_i^p) - \boldsymbol{f}_Z^\top(\boldsymbol{x}_i, \boldsymbol{\theta}_c, \boldsymbol{t})\ \boldsymbol{\beta}_Z - \boldsymbol{f}_D^\top(\boldsymbol{x}_i, \boldsymbol{\theta}_c, \boldsymbol{t})\ \boldsymbol{\beta}_D$ for $i = 1, 2, \ldots, n_p$. The ith element is $y^s(\boldsymbol{x}_{i-n_p}^s, \boldsymbol{c}_{i-n_p}, \boldsymbol{t}_{i-n_p}) - \boldsymbol{f}_Z^\top(\boldsymbol{x}_{i-n_p}, \boldsymbol{c}_{i-n_p}, \boldsymbol{t}_{i-n_p})\ \boldsymbol{\beta}_Z$ for $i = n_p + 1, n_p + 2, \ldots, n_p + n_s$.

It is worth noting that the two terms in (4.10) reflect the intuition of simultaneous tuning and calibration. A large value of $[E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\ \boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s,\ \boldsymbol{y}^p, \boldsymbol{t})]^2$ implies a

large discrepancy between the computer experiment and the physical experiment. So $[E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})]^2$ can be interpreted as *squared prediction bias*. The term $Var(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})$ is a measure of the uncertainty of $\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})$ given $(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})$. Intuitively, our methodology favors a $\boldsymbol{t}$ that minimizes the sum of the squared prediction bias and the uncertainty.

Given that $E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}) = E_{[\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]}[E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})]$ and that $[\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]$ can be simulated by the Metropolis-Hastings algorithm, the Law of Large Numbers guarantees

$$E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}))^2|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}) \approx \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E(\Delta^2(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \boldsymbol{t})|\widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}), \qquad (4.12)$$

where $(\widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l)$ is a sequence of (approximately independent) draws of $(\boldsymbol{\theta}_c, \boldsymbol{\phi})$ from $[\boldsymbol{\theta}, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]$ and N_{MC} is a sufficiently large number. Thus, we can asymptotically approximate $E(\Delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})$.

Second, we estimate $\boldsymbol{t}^\star$ by

$$\begin{aligned} \widehat{\boldsymbol{t}^\star} &= \underset{\boldsymbol{t}}{\text{argmin}} \int_0^1 \int_0^1 E[(\Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}))^2|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}][\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}] d\boldsymbol{\theta}_c d\boldsymbol{x} \\ &\approx \underset{\boldsymbol{t}}{\text{argmin}} \int_0^1 \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E[(\Delta(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \boldsymbol{t}))^2|\widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l, \boldsymbol{Y}^s, \boldsymbol{Y}^p, \boldsymbol{t}] d\boldsymbol{x}. \qquad (4.13) \end{aligned}$$

The integral over $\boldsymbol{x}$ in (4.13) can be approximated by a Monte Carlo integration that averages (4.12) over a grid of $\boldsymbol{x}$ inputs. In our examples, we took

$$\widehat{\boldsymbol{t}^\star} \approx \underset{\boldsymbol{t}}{\text{argmin}} \frac{1}{N_x} \frac{1}{N_{MC}} \sum_{i=1}^{N_x} \sum_{l=1}^{N_{MC}} E[(\Delta(\boldsymbol{x}_i, \widehat{\boldsymbol{\theta}}_{c,l}, \boldsymbol{t}))^2|\widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}], \qquad (4.14)$$

where $\{\boldsymbol{x}_i \in [0,1]^{p_x} | i = 1, \ldots, N_x\}$ are points taken to approximate the integration in (4.13). In the examples we consider, we use an equally-spaced grid of values of $\boldsymbol{x}$ for the Monte Carlo integration and take $N_x = 101$. By the Law of Large numbers,

$\widehat{\boldsymbol{t}}^{\star}$ converges to $\boldsymbol{t}^{\star}$ almost everywhere as N_x and N_{MC} go to infinity. We regard $\widehat{\boldsymbol{t}}^{\star}$ in (4.14) as the estimator of $\boldsymbol{t}^{\star}$ and the quantity $\sum_{i=1}^{N_x}\sum_{l=1}^{N_{MC}} E[(\Delta(\boldsymbol{x}_i, \widehat{\boldsymbol{\theta}}_{c,l}, \boldsymbol{t}))^2|\, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]/(N_x N_{MC})$ as the *estimated squared discrepancy.*

Third, we estimate the posterior distribution $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}^{\star}]$. Because $\widehat{\boldsymbol{t}}^{\star}$ converges to $\boldsymbol{t}^{\star}$ almost everywhere, $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}]$ converges to $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}^{\star}]$ in distribution. We therefore estimate the distribution $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}^{\star}]$ using the draws of $\boldsymbol{\theta}_c$ from $[\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}]$.

Operationally, we conduct simultaneous tuning and calibration as follows.

Step 1 For each possible $\boldsymbol{t}$ in a grid of the tuning parameter vectors, make draws $\{(\widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l); l = 1, \ldots, N_{MC}\}$ from $[\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]$ and compute the estimated squared discrepancy at a set of inputs $\{\boldsymbol{x}_i; i = 1, \ldots, N_x\}$.

Step 2 Compute $\widehat{\boldsymbol{t}}^{\star} = \text{argmin}_{\boldsymbol{t}} \frac{1}{N_x}\frac{1}{N_{MC}} \sum_{i=1}^{N_x}\sum_{l=1}^{N_{MC}} E(\Delta^2(\boldsymbol{x}_i, \widehat{\boldsymbol{\theta}}_{c,l}, \boldsymbol{t})|\, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l, \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t})$.

Step 3 Estimate $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}]$.

We call this the STaC procedure, which stands for Simultaneous Tuning and Calibration.

4.3.3 Prediction

In addition to tuning and calibration, we can predict $y^s(\cdot)$ and $\eta(\cdot)$ and construct the predictive intervals using $\widehat{\boldsymbol{t}}^{\star}$ and draws $\{(\widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l)\}$ from $[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}]$. Given $\widehat{\boldsymbol{t}}^{\star}$, the BLUP of $y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ is $E(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\ |\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}) = E_{\left[\boldsymbol{\theta}_c, \boldsymbol{\phi}|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}\right]}$ $[E(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}, \boldsymbol{\theta}_c, \boldsymbol{\phi})]$. By the Law of Large Numbers,

$$\widehat{y}^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) \approx \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}}^{\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l). \tag{4.15}$$

Similarly, the predictor of $\eta(\boldsymbol{x}, \boldsymbol{\theta}_c)$ is

$$\widehat{\eta}(\boldsymbol{x}, \boldsymbol{\theta}_c) \approx \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E(Y^s(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{t}^\star}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l). \tag{4.16}$$

The uncertainty in the predicted $y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ can be approximated by

$$\begin{aligned}
& Var(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}) \\
= \; & E_{\left[\boldsymbol{\theta}_c, \boldsymbol{\phi} | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}\right]} \left[Var(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \boldsymbol{\theta}_c, \boldsymbol{\phi}) \right] \\
& + Var_{\left[\boldsymbol{\theta}_c, \boldsymbol{\phi} | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}\right]} \left[E(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \boldsymbol{\theta}_c, \boldsymbol{\phi}) \right] \\
\approx \; & \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} Var(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l) \\
& + \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} \left[E(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l) \right]^2 \\
& - \left[\frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l) \right]^2 .
\end{aligned} \tag{4.17}$$

By the similar arguments, the uncertainty in the predicted $\eta(\boldsymbol{x}, \boldsymbol{\theta}_c)$ is the conditional variance of $Y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star})$. This quantity can be approximated by

$$\begin{aligned}
& Var(Y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}) \\
\approx \; & \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} Var(Y^s(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{t}^\star}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l) \\
& + \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} \left[E(Y^s(\boldsymbol{x}, \widehat{\boldsymbol{\theta}_{c,l}}, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{t}^\star}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l) \right]^2 \\
& - \left[\frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E(Y^s(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{t}^\star}) | \boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l}, \widehat{\boldsymbol{\phi}}_l) \right]^2 .
\end{aligned} \tag{4.18}$$

Next we derive $E(Y^s(\boldsymbol{x}, \boldsymbol{c},\ \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p,\ \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi})$, $E(Y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})\ + \Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) |\ \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c,\ \boldsymbol{\phi})$, $Var(Y^s(\boldsymbol{x}, \boldsymbol{c},\ \boldsymbol{t}) | \boldsymbol{y}^s, \boldsymbol{y}^p,\ \boldsymbol{t}, \boldsymbol{\theta}_c,\ \boldsymbol{\phi})$, and $Var(Y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})\ + \Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t}) |\ \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi})$. Using (4.2), (4.3), (4.4), (4.5), (4.6), and the following results, one can compute (4.15), (4.16), (4.17), and (4.18) by taking $(\boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi})$ to be $(\widehat{\boldsymbol{t}^\star}, \widehat{\boldsymbol{\theta}}_{c,l},\ \widehat{\boldsymbol{\phi}}_l)$.

- The conditional expectation of $Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ is $E(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p,\ \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi}) = \boldsymbol{f}_Z^\top(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})\ \boldsymbol{\beta}_Z + \boldsymbol{\Sigma}_{0Z}^\top \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{D}_{\boldsymbol{t}}$, where $\boldsymbol{\Sigma}_{0Z}$ is an $(n_p + n_s) \times 1$ vector of covariances between $Z(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ and the entries of $(\boldsymbol{Y}^{p\top}, \boldsymbol{Y}^{s\top})^\top$. The ith element of $\boldsymbol{\Sigma}_{0Z}$ is $Cov(Z(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}),\ Z(\boldsymbol{x}_i^p,\ \boldsymbol{\theta}_c, \boldsymbol{t})|(\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}))$ if $i = 1, \ldots, n_p$. The ith element is $Cov(Z(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}),\ Z(\boldsymbol{x}_{i-n_p}^s, \boldsymbol{c}_{i-n_p}, \boldsymbol{t}_{i-n_p})|\ (\boldsymbol{\theta}_c, \boldsymbol{\phi}, \boldsymbol{t}))$ for $i = n_p + 1, \ldots, n_p + n_s$.

- The conditional variance of $Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ is $Var(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi}) = \sigma_Z^2 - \boldsymbol{\Sigma}_{0Z}^\top \boldsymbol{\Sigma}_{ZZ}^{-1} \boldsymbol{\Sigma}_{0Z}$.

- The conditional mean of $\eta(\boldsymbol{x}, \boldsymbol{\theta}_c)$ is $E(Y^s(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi}) + E(\Delta(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})|\ \boldsymbol{y}^s, \boldsymbol{y}^p,\ \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi})$.

- The conditional variance of $Y^s(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t}) + \Delta(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})$ is $Var(Y^s(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t}) + \Delta(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{t})|\ \boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}, \boldsymbol{\theta}_c, \boldsymbol{\phi}) = \sigma_Z^2 + \sigma_D^2 - (\boldsymbol{\Sigma}_{0Z}^\top + \boldsymbol{\Sigma}_{DZ}^\top) \boldsymbol{\Sigma}_{ZZ}^{-1} (\boldsymbol{\Sigma}_{0Z} + \boldsymbol{\Sigma}_{DZ}^\top)$.

Based on (4.15), (4.16), (4.17), (4.18), and the results listed above, we can compute point-wise $100(1-\alpha)\%$ prediction bands for $y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})$ and $\eta(\boldsymbol{x}, \boldsymbol{\theta}_c)$ as

$$\widehat{y}^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t}) \pm z^{\alpha/2} \sqrt{Var(Y^s(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{t})|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star})}$$

and

$$\widehat{\eta}(\boldsymbol{x}, \boldsymbol{\theta}_c) \pm z^{\alpha/2} \sqrt{Var(Y^s(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star}) + \Delta(\boldsymbol{x}, \boldsymbol{\theta}_c, \widehat{\boldsymbol{t}^\star})|\boldsymbol{y}^s, \boldsymbol{y}^p, \widehat{\boldsymbol{t}^\star})},$$

where $z^{\alpha/2}$ is the upper $\alpha/2$ critical point of the standard normal distribution.

4.4 Examples and Comparison

4.4.1 Discussion

In this section, the `gpmsa` program is compared with STaC. Bayesian calibration requires setting a prior distribution on the tuning parameters as well as the calibration parameters. On the other hand, STaC minimizes a discrepancy between the

computer and the physical experiment output to set the tuning parameters and samples an appropriate posterior distribution to assess calibration parameters. Thus the necessary and sufficient condition that Bayesian calibration and STaC are equivalent is that the marginal posterior distribution of $\boldsymbol{t}$ and the discrepancy measure give the same estimation of the tuning parameters. Lemma 1 in Loeppky et al. (2006) shows that, roughly, for their Gaussian stochastic process model, under the conditions that

(a) the discrepancy between the computer and the physical experiments can be minimized to be zero and

(b) infinitely many runs from the computer simulation and the physical experiment are available,

the MLE of the discrepancy converges to zero as the number of the computer runs goes to infinity and the MLE of the calibration parameters can minimize the discrepancy asymptotically. Based on their Lemma 1, Bayesian calibration and STaC can give comparable estimates of $\boldsymbol{t}^\star$ in our model if

$(a^\star)$ by setting the calibration (or tuning) parameters to certain values, the discrepancy $\delta(\cdot)$ can be minimized sufficiently close to zero,

$(b^\star)$ the number of the observations from the computer experiment is reasonably large, and

$(c^\star)$ the posterior mode of every calibration parameter is close to the MLE of the parameter.

However, $(a^\star)$, $(b^\star)$, and $(c^\star)$ can be hard to verify (or false) in many applications, which is caused by three major reasons. The discrepancy $\delta(\cdot)$ need not be minimized

to zero because of the complexity of the real world phenomena and the inadequacy of the computer simulation code. The numbers of the computer and physical experiment runs are typically limited. The posterior mode of a calibration parameter can differ from the MLE if either the prior for $\boldsymbol{\theta}_c$ or the prior for $\Delta(\cdot)$ is informative. Therefore, the Bayesian calibration may fail to minimize the difference between the computer and the physical experiments.

4.4.2 An Illustrative Example with Known $t^\star$ and θ_c

In this example there are four real-valued inputs to a computer code: a control variable x, one tuning parameter t, and two calibration parameters c_1 and c_2. All the four inputs have support $[0, 1]$. The output from the computer code was generated as

$$y^s(x, c_1, c_2, t) = c_1 e^{-c_2 x} + 10 \times (t - 0.5)^2$$

and the response from the physical experiment was generated as

$$y^p(x) = \eta(x, \boldsymbol{\theta}_c) + \epsilon(x) = e^{-x} + ((x - 0.5)^2 - 0.125) + \epsilon(x),$$

where $y^s(\cdot)$ and $y^p(\cdot)$ denote the outputs from the computer code and the physical experiment and $\epsilon(x)$ denotes the white noise Gaussian random error having mean 0 and standard deviation 0.01. In this example the bias term is

$$e^{-x}(1 - C_1 e^{x - c_2 x}) + x^2 - x - 10t^2 + 10t - 2.375,$$

which can not be driven to zero by any setting of (c_1, c_2, t). However, it can be checked that by setting $c_1 = c_2 = 1$, $y^s(x, 1, 1, t)$ and $y^p(x)$ have the same exponential part for all t and that $(c_1, c_2, t) = (0.94, 1.0, 0.5)$ minimizes the L_2 discrepancy $\int_0^1 (e^{-x} + ((x-0.5)^2 - 0.125) - y^s(x, c_1, c_2, t))^2 \, dx$. (The minimizers of L_1 and L_∞ discrepancies

are $(c_1, c_2, t) = (0.91, 1.0, 0.5)$ and $(c_1, c_2, t) = (1.0, 1.0, 0.5)$.) We therefore desire the estimate of t to be close to 0.5 and the posterior distributions of c_1 and c_2 to be concentrated on values near 1.

We generated 50 inputs (x, c_1, c_2, t) for the computer experiment and 20 inputs x for the physical experiment using Maximin Latin Hypercube Designs. The Bayesian calibration program `gpmsa` (Gattiker (2005)) was run treating c_1, c_2, and t as calibration parameters with unknown true values $\theta_{c_1}, \theta_{c_2}$, and θ_t. Following 8000 burn-in iterations and 2000 production ones, we regarded 100 equally spaced samples drawn from the 2000 production runs as independent draws from the joint posterior distribution of $(\theta_{c_1}, \theta_{c_2}, \theta_t)$. The estimated posterior distributions of θ_t, θ_{c_1}, and θ_{c_2} are shown in Figure 4.3. The estimated $[\theta_t | \boldsymbol{y}^p, \boldsymbol{y}^s]$ fails to pinpoint that $t = 0.5$ and neither do the estimated $[\theta_{c_1} | \boldsymbol{y}^p, \boldsymbol{y}^s]$ and $[\theta_{c_2} | \boldsymbol{y}^p, \boldsymbol{y}^s]$ clearly suggest large values. Thus, determining the tuning and calibration parameters is difficult for this data. We next

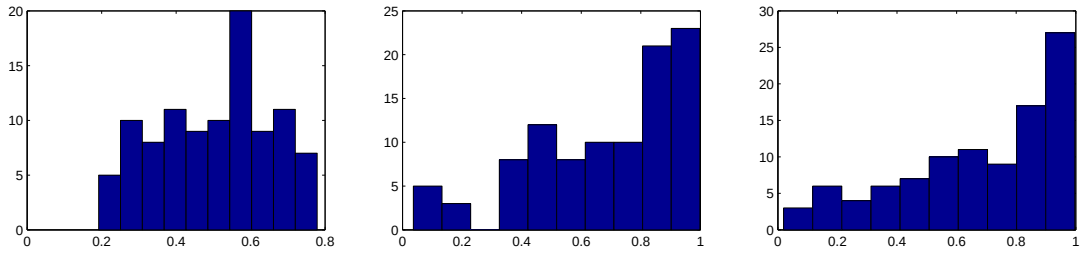

Figure 4.3: Simulated posterior distributions of θ_t (the left panel), θ_{c_1} (the middle panel), and θ_{c_2} (the right panel) using the Bayesian calibration program.

used `gpmsa` to predict $\eta(x, \boldsymbol{\theta}_c) = \eta(x, \theta_{c_1}, \theta_{c_2})$ over a grid of equally-spaced inputs; i.e., $x = 0, 0.02, \ldots, 1$. Figure 4.4 depicts the training data, the true response curve, and

the predictions. Although the predictions are quite accurate when $x \in [0.2, 0.6]$, the predictions are biased when x is close 0 or 1. We compute the RMSPE as a measure of the predictive accuracy. The RMSPE of a generic predictor $\widehat{\eta}(\cdot)$ is defined as

$$\text{RMSPE}(\widehat{\eta}) = \sqrt{\frac{1}{51}\sum_{i=1}^{51}(\eta(x_i, \boldsymbol{\theta}_c) - \widehat{\eta}(x_i, \boldsymbol{\theta}_c))^2}.$$

The RMSPE of the predictor obtained by `gpmsa` is 0.1662.

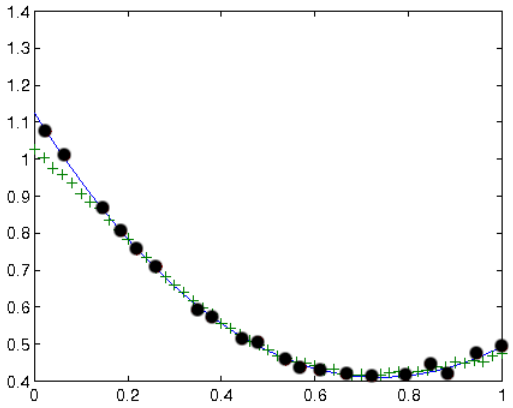

Figure 4.4: The training data (solid circles), the true response curve (the solid line), and the predictions (pluses) obtained by `gpmsa`.

Next we apply STaC to this example with $N_{MC} = 100$ and $N_x = 101$. The number of iterations in the burn-in period, the number of production runs, the sampling space, and the prediction inputs are the same as used for `gpmsa`. Table 4.2 lists the L_2 discrepancy for a grid of t and the corresponding estimated squared discrepancies. Thus, STaC picks $\widehat{t^\star} = 0.5$. We then simulated the posterior distributions of θ_{c_1} and θ_{c_2}. Figure 4.5 shows histograms of the simulated posterior distributions of θ_{c_1} and θ_{c_2}; both posteriors are concentrated on values close to one.

t	0.1	0.2	0.3	0.4	**0.5**	0.6	0.7	0.8	0.9
Equation (4.14)	7.85	16.62	14.27	3.64	0.83	21.42	29.50	65.90	14.32

Table 4.2: Grid of t and the approximate integral (4.14)

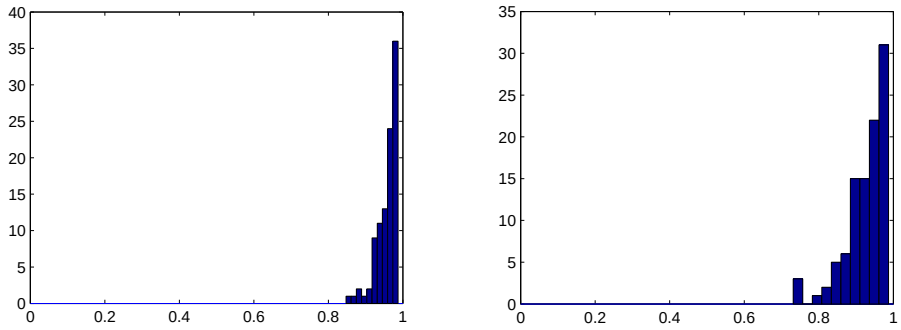

Figure 4.5: Simulated posterior distributions of θ_{c_1} (the left panel) and θ_{c_2} (the right panel) using STaC

Figure 4.6 plots the predictions and the 99% two-sided prediction bands for true input-output function $\eta(x, \boldsymbol{\theta}_c)$ using STaC. We see that the predictions are close to the true responses and the 99% prediction band contains the true curve for all $x \in [0, 1]$. The RMSPE of the predictor obtained by the STaC program is 0.0445. The relative improvement of STaC compared with the `gpmsa`-based predictor is 73.23% $(= (0.1662 - 0.0445)/0.1662 \times 100\%)$.

We conclude that for this example, the inferences of the tuning and calibration parameters and the predictions with STaC are more informative and more accurate than using Bayesian calibration.

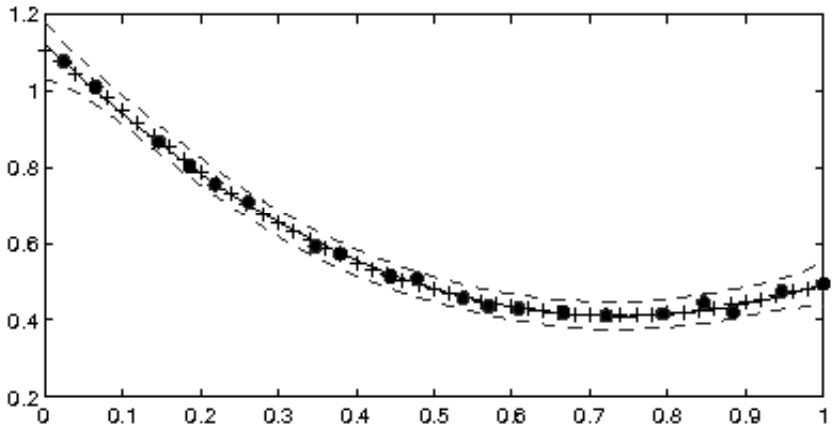

Figure 4.6: The training data (solid circles), true response curve (the solid line), predictions (pluses), and 99% prediction bands (dashes) using the STaC program.

4.4.3 A Biomechanics Example

We apply STaC to the biomechanics application sketched in Section 2. The number of burn-in and production runs are identical to those used in our first example. The left panel of Figure 4.7 plots the estimated squared discrepancies for the tuning parameter equal to $15, 16, \ldots, 30$ (with $N_{MC} = 100$ and $N_x = 101$); this figure shows that the computer simulation best matches the knee simulator when the load discretization is set to $\widehat{t^\star} = 19$. The right panel shows the histogram of the simulated posterior distribution $[\theta_c | \boldsymbol{Y}^p, \boldsymbol{Y}^s, \widehat{t^\star} = 19]$. We can see from Figure 4.7 that the simulated posterior distribution of initial position has mode -1.7 with substantially smaller uncertainty than the Bayesian calibration program (`gpmsa`). These results can be used for future runs of the FEA code to study the anterior-posterior displacement.

Figure 4.8 depicts the predictions and 99% prediction bands for the anterior-posterior displacement in the physical experiment using the current data. Once initial

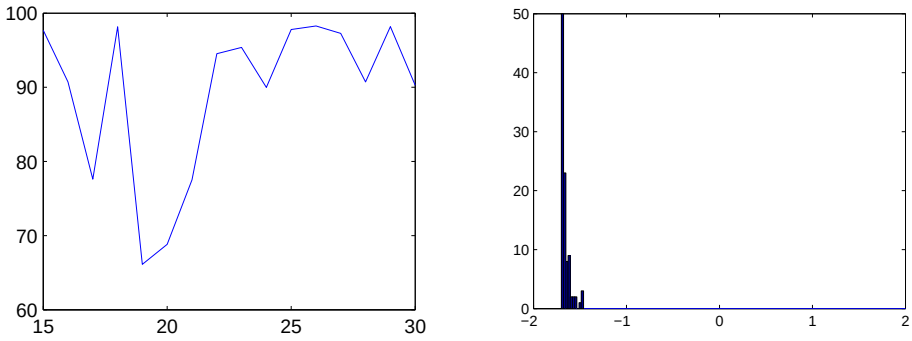

Figure 4.7: A plot of the estimated squared discrepancy against the value of the tuning parameter (the left panel) and a histogram of the simulated posterior distribution of the calibration parameter (the right panel).

position and load discretization are set using STaC, the predictions clearly match those provided by the knee simulator.

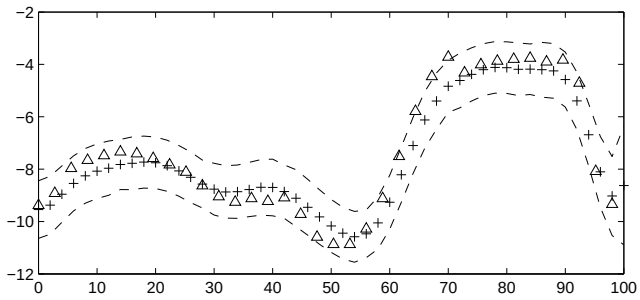

Figure 4.8: The training data (triangles), predictions (pluses), and 99% prediction bands (dashes) using the STaC program.

4.5 Summary and Future Research

In this chapter, we introduce a Bayesian methodology for simultaneous tuning and calibration and demonstrate, with examples, that STaC performs well for tuning, calibration, and prediction of the true input-output relationship. When the bias in the computer experiment output cannot be reduced to zero by setting the calibration parameter equal to its true value, one should treat tuning and calibration parameters differentially. Given this is typically the case, we recommend that STaC is a conservative method for setting parameters. In particular, we recommend that tuning parameters should be set using a discrepancy measure. The conditional distribution of $\boldsymbol{\theta}_c$ given the data and the best choice of the tuning parameters should be used to make inference about the calibration parameters.

Research is continuing on three topics concerning STaC. Currently, determining the tuning parameters in the first step of STaC can be time consuming; additional work on speeding up the computation is needed. Second, a measure of the uncertainty in the estimated tuning parameter $\widehat{\boldsymbol{t}}^{\star}$, perhaps based on the curvature of the integral $\int_0^1 \int_0^1 \delta^2(\boldsymbol{x}, \boldsymbol{\theta}_c, \boldsymbol{t})[\boldsymbol{\theta}_c|\boldsymbol{y}^s, \boldsymbol{y}^p, \boldsymbol{t}]d\boldsymbol{\theta}_c d\boldsymbol{x}$, is needed. The third topic is more long-range. It would be of interest to extend STaC to other settings, for example, to cases where outputs are multivariate and to applications where both the computer output and the physical experiment have mixed quantitative and qualitative inputs.

BIBLIOGRAPHY

Adler, R. J. (1981). *The Geometry of Random Fields*. J. Wiley, New York.

Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., Sacks, J. and Walsh, D. (2007a). Computer Model Validation with Functional Output. *The Annals of Statistics* **35**(5), 1874–1906.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H. and Tu, J. (2007b). A Framework for Validation of Computer Models. *Technometrics* **49**(2), 138–154.

Christiansen, C. and Morris, C. N. (1997). Hierarchical Poission regression modeling. *Journal of the American Statistical Association* **92**, 618–632.

Conti, S. and Hagan, A. O. (2006). Bayesian Emulation of Complex Multi-Output and Dynamic Computer Models. *Technical report*. Department of Probability and Statistics, University of Sheffield.

Cox, D. D., Park, J. S. and Singer, C. E. (1996). A statistical method for tuning a computer code to a database. *Technical Report 96-3*. Department of Statistics, Rice University.

Craig, P. C., Goldstein, M., Rougier, J. C. and Seheult, A. H. (2001). Bayesian Forecasting for Complex Systems using Computer Simulators. *Journal of the American Statistical Association* **96**, 717–729.

Craig, P. S., Goldstein, M., Seheult, A. H. and Smith, J. A. (1996). Bayes linear strategies for history matching of hydrocarbon reservoirs. In *Bayesian Statistics*, Vol. 5 (J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (eds)), pp. 69–95, Oxford University Press.

Cressie, N. A. (1993). *Statistics for Spatial Data*. J. Wiley, New York.

Currin, C., Mitchell, T. J., Morris, M. D. and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association* **86**, 953–963.

Dean, A. M. and Voss, D. (1999). *Design and Analysis of Experiments*. Spring Verlag, New York.

Fang, K. T., Li, R. and Sudjianto, A. (2005). *Design and Modeling for Computer Experiments*. Chapman and Hall.

Gattiker, D. H. J., Williams, B. and Rightley, M. (2005). Computer Model Calibration using High Dimensional Output. *Technical Report LA-UR 05-6410*. Los Alamos National Laboratory.

Gattiker, J. (2005). Using the Gaussian Process Model for Simulation Analysis (GPM/SA) Code. *Technical report*. Los Alamos National Laboratory.

Higdon, D., Kennedy, M., Cavendish, J., Cafeo, J. and Ryne, R. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal of Scientific Computing* **26**, 448–466.

Higdon, D., Williams, B., Moore, L., McKay, M. and Keller-McNulty, S. (2005). Uncertainty Quantification for Combining Experimental Data and Computer Simulations. *Technical Report LA-UR 05-4951*. Los Alamos National Laboratory.

Johnson, M. E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26**, 131–148.

Joseph, V. R. (2006). Limit Kriging. *Technometrics* **48**, 458–466.

Joseph, V. R., Hung, Y. and Sudjianto, A. (2007). Blind Kriging: A New Method for Developing Metamodels. *Technical Report 80-2172*. School of Industrial and Systems Engineering, Georgia Institute of Technology.

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

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

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D. and Ye, K. Q. (2006). Variable Selection for Gaussian Process Models in Computer Experiments. *Technometrics* **48**, 478–490.

Loeppky, J., Bingham, D. and Welch, W. (2006). Computer Model Calibration or Tuning in Practice. *Technical report*. University of British Columbia.

Long, J. P. (2008). *Parametric analyses of hip resurfacing femoral components: influence of design, environmental, and surgical variables*. PhD thesis. Cornell University. 127 pages.

Long, J. P. and Bartel, D. L. (2006). Surgical Variables Affect the Mechanics of a Hip Resurfacing System. *Clinical orthopaedics and related research* **453**, 115–122.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology* **58**, 1246–1266.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245.

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

Morris, M. D., Mitchell, T. J. and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics* **35**, 243–255.

Park, J. S. (1991). *Tuning Complex Computer Codes to Data and Optimal Designs*. PhD thesis. University of Illinois. Champaign/Urbana, IL USA.

Parzen, E. (1967). *Statistical Inference on Time Series by Hilbert Space Methods I. In: Time Series Analysis Papers by Emanuel Parzen, 251-382*, San Francisco.

Patterson, H. D. and Thompson, R. (1971). Recovery of interblock information when block sizes are unequal. *Biometrika* **58**, 545–554.

Qian, P. Z. G. and Wu, C. F. J. (2008). Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments. *Technometrics* **50**(2), 192–204.

Qian, Z., Wu, H. and Wu, J. (2008). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* **50**(3), 383–396.

Rawlinson, J. J., Furman, B. D., Li, S., Wright, T. M. and Bartel, D. L. (2006). Retrieval, Experimental, and Computational Assessment of the Performance of Total Knee Replacements. *Journal of Orthopaedic Research* **24(7)**, 1384–1394.

Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer-Verlag.

Sacks, J., Schiller, S. B. and Welch, W. J. (1989a). Design for computer experiments. *Technometrics* **31**, 41–47.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989b). Design and analysis of computer experiments. *Statistical Science* **4**, 409–423.

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

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

Tang, B. (1993). Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association* **88**, 1392–1397.

Wallstrom, T. (2007). Estimating Replicate Variation. *Technical Report T-13*. Los Alamos National Laboratory.

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