BAYESIAN FUNCTIONAL DATA ANALYSIS FOR COMPUTER MODEL VALIDATION

by

Fei Liu

Department of Institute of Statistics and Decision Sciences
Duke University

Date: _____

Approved:

_____
Dr. James Berger, Supervisor

_____
Dr. Susie Bayarri

_____
Dr. Merlise Clyde

_____
Dr. Feng Liang

_____

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Institute of Statistics and Decision Sciences
in the Graduate School of
Duke University

May 2007

ABSTRACT

BAYESIAN FUNCTIONAL DATA ANALYSIS FOR COMPUTER MODEL VALIDATION

by

Fei Liu

Department of Institute of Statistics and Decision Sciences
Duke University

Date: _____

Approved:

_____
Dr. James Berger, Supervisor

_____
Dr. Susie Bayarri

_____
Dr. Merlise Clyde

_____
Dr. Feng Liang

_____

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Institute of Statistics and Decision Sciences in the Graduate School of Duke University

May 2007

Abstract

Functional data analysis (FDA) – inference on curves or functions – has wide application in statistics. An example of considerable recent interest arises when considering computer models of processes; the output of such models is a function over the space of inputs of the computer model. The output is functional data in many contexts, such as when the output is a function of time, a surface, etc. In this research, we develop or extend four Bayesian FDA approaches to computer model validation, tailored to interdisciplinary problems in engineering and the environment.

The first approach we consider is a nonparametric Bayesian statistics approach, utilizing a separable Gaussian Stochastic Process as the prior distribution for functions. This is a natural choice for smooth functions. The methodology is applied to a thermal computer model challenge problem, proposed by the `Sandia National Laboratory`.

Direct use of separable Gaussian stochastic processes is inadequate for irregular functions, and can be computationally infeasible for high dimensional functions. The approach developed for such functions consists of representing the function in the wavelet domain; reducing the number of nonzero coefficients by thresholding; modeling the nonzero coefficients as functions of the associated inputs, using the nonparametric Bayesian method; and reconstructing the functions (with confidence bands) in the original (time) domain.

The third approach extends the second in terms of function representation. We represent the functions in the eigen-space whose basis elements are linear combinations of the wavelet basis elements. The number of nonzero coefficients is

greatly reduced in this eigen-space, as consequently is the computational expense for the statistical inverse problem. This method is applied to computer modeling of vehicle suspension systems.

The fourth approach models functions as multivariate Dynamic Linear Models. This approach is useful when the functions are highly variable and, as opposed to attempting to represent the functions exactly, one seeks primarily to capture relevant stochastic structure of the functions. The method has been tested with a simulated data set.

In addition to the basic issue of functional data, all the above approaches must also contend with three other issues associated with computer model validation. First, emulators must typically be constructed for expensive-to-run computer models, by treating them as spatial processes defined on the input space. Second, computer model bias – the discrepancy between the computer model output and reality – must be taken into account. Third, the computer models typically have unknown parameters, requiring solution of an inverse problem in their estimation. Because these issues must all be addressed simultaneously and with limited data, extensive use is made of Markov Chain Monte Carlo (MCMC) algorithms. Some modular versions of MCMC are also introduced to reduce the confounding between some of the elements in the corresponding statistical models.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to acknowledge with gratitude my friends and colleagues who have helped and supported me over the past five years. My primary thanks go to my advisor, Dr. James Berger, for his constant support and motivating encouragement during the development of this research and throughout my experience of graduate study. It has been a privilege and a honor to learn about statistics and appreciate the beauty of science by working with him. His enthusiasm for and commitment to research work set a strong example that I hope to follow.

I would like to thank Dr. Susie Bayarri, Dr. Rui Paulo, and Dr. Jerry Sacks for their important roles as collaborators in this research. Without their help and support, parts of this dissertation would be entirely impossible. I would also like to thank Dr. Merlise Clyde, for the many valuable discussions on wavelets, and Dr. Mike West, for his innumerable discussions on Dynamic Linear Models. I specially thank Dr. Feng Liang for the academic advice and personal support she has offered me throughout these years.

I am grateful to all people at Duke University and the Statistical and Applied Mathematical Sciences Institute. Thanks to Floyd Bullard, Satkartar Kinney, Simon Lunagomez, Natesh Pillai, Gavino Puggioni, Abel Rodriguez, Huiyan Sang, for sharing with me every day in Old Chem 222. Thanks Lance Brown, Elizabeth Clark, Patricia Johnson, Krista Moyle, Donna Pike, Nicole Scott for their administrative support. Special thanks to Zhi Ouyang and Ying Yang, for the great help they offered me during my crazy interviews.

I dedicate this dissertation to my family, Mom, Dad, and my husband Yuhong Wu for their endless support, love and encouragement.

Chapter 1

Introduction

Computer models have become essential tools for describing complex physical processes in many scientific disciplines. Applications range from modeling climate to engineering new technology. Computer models in such areas are essential because physical experiments may be too expensive or simply be impossible to carry out.

The computer model takes input and returns output. Typically, the model is deterministic – it produces identical output values with the same inputs. We view the model as a function of the input $\boldsymbol{z}$, denoted by $y^M(\boldsymbol{z})$. Statistical analysis of computer models faces the following challenges. First, it is often time consuming to run the computer model. As a result, we can only observe $y^M(\cdot)$ at a limited number of choices for the input. Second, the values for the input variables may be unknown for the real physical process. Field runs, data obtained by conducting real physical experiments, are often used to calibrate those unknowns. Third, the computer models are never completely accurate representations of the real processes being modeled. Statistical analysis should be able to incorporate the model inadequacy.

This dissertation is motivated by the methodological need to handle irregular functional outputs produced by computer models and uncertainty in the computer model inputs. Within the general framework of Bayesian statistics, this dissertation develops four approaches, each illustrated by a specific application.

1.1 Statistical Analysis of Computer Models

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

Designing computer experiments - The goal is to choose $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, the input values at which the computer model will be exercised. McKay *et al.* (1979) first introduced Latin hypercube sampling to design computer experiments. The intuition is to cover the range of the key input values and fill the space effectively (Sacks *et al.* (1989); Bates *et al.* (1996)). In the examples we consider here, the design points are selected by a Latin Hypercube Design that maximizes the minimum distance between the design points. More precisely, we choose $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ subject to

$$\max_{LHD} \min_{i,j} \mathrm{d}(\boldsymbol{z}_i, \boldsymbol{z}_j)\,,$$

where $\mathrm{d}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is the distance between design points $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$.

Uncertainty and sensitivity analysis - The problem of *uncertainty analysis* is to study the distribution of the output induced by the input distributions, and *sensitivity analysis* is more focused on identifying inputs to which the output is relatively sensitive. Saltelli *et al.* (2000) presents the large literature on this problem. See also Santner *et al.* (2003) for a review. Recent research papers include Oakley and O'Hagan (2002), Oakley (2004), Oakley and O'Hagan (2004) and Linkletter *et al.* (2006).

Model the computer model outputs - The objective is to predict the computer model output $y^M(\boldsymbol{z})$, at any untried input $\boldsymbol{z}$, conditional on the computer model runs at the design points $y^M(\boldsymbol{z}_1), \ldots, y^M(\boldsymbol{z}_n)$. The resulting prediction, together with the associated uncertainty, is then used as a fast surrogate to the

computer model, often called an *emulator*. The Gaussian Response Surface Approximation method *(GaSP)*, following on work in Sacks *et al.* (1989), Currin *et al.* (1991), Welch *et al.* (1992), and Morris *et al.* (1993), has received considerable attention in the literature and has become the most common approach to constructing an emulator. The main idea of *GaSP* is to assign a Gaussian process as the prior distribution for y^M, and update our knowledge of the computer model $y^M(\cdot)$ upon observing the evaluations of $y^M(\cdot)$ at inputs $D^M = \{\boldsymbol{z}_i : i = 1, \ldots, n\}$. This yields a fast emulator for y^M at any untried input $\boldsymbol{z}$.

Bayarri *et al.* (2005b) and Bayarri *et al.* (2006) incorporate *GaSP* with functional output. Bayarri *et al.* (2005b) treat time t as another input, and allow hierarchical modeling to combine experimental data from different conditions. Bayarri *et al.* (2006) represent functional data by a basis expansion. A related approach for computer models with high dimensional outputs is in Higdon *et al.* (2007), where the data is represented by principle components.

Calibration - Some of the computer model inputs are typically unknown parameters. The objective of *calibration* is learn about them in conjunction with the field runs. The traditional approach to calibration is to search for the values of the unknown inputs whose corresponding model outputs best fit the field runs. This *ad hoc* search takes no account of the parameter uncertainty and model inadequacy in the subsequent analysis. Kennedy and O'Hagan (2001) classify sources of uncertainty arising in the use of computer models. The Bayesian approach they proposed is the first attempt to explicitly model all the sources of uncertainty. A fully Bayesian version is described in Bayarri *et al.* (2002).

Another Bayesian approach to calibration is given by Craig *et al.* (1997), Craig *et al.* (2001), Goldstein and Rougier (2003), Goldstein and Rougier (2004) and

Goldstein and Rougier (2006), which focus on the use of linear Bayes methodology Goldstein (1998).

Validation - As we have discussed, a computer model is often a biased representation of the real process. The reasonable objective of *validation* is thus to answer the relevant question *"Does the computer model adequately represent reality?"*

Bayarri *et al.* (2005a) described a general framework for validation of complex computer models and applied the framework to two examples involving scalar data. The procedure treats the combined calibration/validation process, and assesses the possible systematic differences between model outcomes and test outcomes (so-termed biases), by estimating these biases along with uncertainty bounds for these estimates. Bayarri *et al.* (2005b) generalized this work to the situation of smooth functional data, arising within a hierarchical structure, and Bayarri *et al.* (2006) generalized to the situation of irregular functional data.

1.2 Computer Model Validation Framework

The Simulator Assessment and Validation Engine (*SAVE*) (Bayarri *et al.*, 2005a) is a Bayesian-based analysis that combines computer simulation results with output from field experiments to produce assessment of the adequacy of a computer model. The method follows these six steps:

(1) Specify the Input/Uncertainty (I/U) map, which consists of prior knowledge on uncertainties or ranges of the computer model inputs and parameters.

(2) Set the evaluation criteria for intended applications.

(3) Collect data – both field and computer runs;

(4) Approximate, if necessary, computer model output;

(5) Compare computer model output with field data using Bayesian statistical analysis;

(6) Feed back the analysis to improve the current validation scheme and computer model, and feed forward to future validation activities.

The central technical issues for *SAVE* lie in implementing Steps (4) and (5). Letting $y^M(\cdot)$ be the computer model output, Step (4) aims at obtaining a fast surrogate of $y^M(\cdot)$, $\hat{y}^M(\cdot)$, that is easy to evaluate. The Gaussian Response Surface Approximation method *(GaSP)*, following on work in Sacks *et al.* (1989), Currin *et al.* (1991), Welch *et al.* (1992), and Morris *et al.* (1993), is the key component for implementing Step (4) for scalar output. The main idea of *GaSP* is to assign a Gaussian process as the prior distribution for y^M,

$$y^M \sim \mathrm{GP}\left(\boldsymbol{\Psi}'(\cdot)\boldsymbol{\theta}^L, \frac{1}{\lambda^M}c^M(\cdot,\cdot)\right),$$

where $\boldsymbol{\Psi}(\cdot)$ is a vector of known functions, $\boldsymbol{\theta}^L$ is a vector of unknown regression coefficients, λ^M is the precision (inverse of variance) of the Gaussian process, and $c^M(\cdot,\cdot)$ is the correlation function. Denoting the arguments/inputs to the computer model by $\boldsymbol{z}$, we assume that $c^M(\cdot,\cdot)$ has the form

$$c^M(\boldsymbol{z},\boldsymbol{z}^*) = \exp\left(-\sum_{j=1}^{d}\beta_j^M \mid \boldsymbol{z}_j - \boldsymbol{z}_j^* \mid^{\alpha_j^M}\right).$$

In the definition of the correlation function, d is the number of coordinates in $\boldsymbol{z}$; the β_j^M's are range parameters controlling the decay of the spatial correlations, $\beta_j^M \geq 0$; the $\alpha_j^M \in [1,2]$ are called roughness parameters which reflect the smoothness of the realizations from the Gaussian process – the function is infinitely differentiable if and only if $\alpha_j^M = 2$ for all j. For simplicity, we denote the hyper-parameters by

$\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\theta}^L, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$.

The correlation structure in the GP prior implies that close-by values of inputs lead to high correlations and hence similarity of outputs, while far apart inputs lead to near zero correlations and a lack of similarity between the outputs. These are typical features of smooth functions, the kind expected to come from many computer models in engineering.

Our knowledge of the computer model $y^M(\cdot)$ will be updated upon observing the evaluations of $y^M(\cdot)$ at inputs $D^M = \{\boldsymbol{z}_i : i = 1, \ldots, m\}$. Let $\boldsymbol{y}^M = (y^M(\boldsymbol{z}_1), \ldots, y^M(\boldsymbol{z}_m))$ be the outputs of the model at D^M. The conditional posterior distribution of y^M given the hyper-parameters, is a Gaussian process with updated mean and covariance function given by

$$\mathbb{E}(y^M(\boldsymbol{z}) \mid \boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M) = \boldsymbol{\Psi}'(\boldsymbol{z})\boldsymbol{\theta}^L + \boldsymbol{\gamma}_z'(\boldsymbol{\Gamma}^M)^{-1}(\boldsymbol{y}^M - \boldsymbol{X}\boldsymbol{\theta}^L)$$

$$\mathbb{C}\mathrm{ov}(y^M(\boldsymbol{z}), y^M(\boldsymbol{z}^*) \mid \boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M) = \frac{1}{\lambda^M} c^M(\boldsymbol{z}, \boldsymbol{z}^*) - \boldsymbol{\gamma}_z'(\boldsymbol{\Gamma}^M)^{-1}\boldsymbol{\gamma}_{z^*}\,,$$

where $\gamma_z' = (c^M(\boldsymbol{z}, \boldsymbol{z}_1), \ldots, c^M(\boldsymbol{z}, \boldsymbol{z}_m))$, $\boldsymbol{\Gamma}^M = (c^M(\boldsymbol{z}_i, \boldsymbol{z}_j))_{i,j}$, and $\boldsymbol{X}$ is the matrix with rows $\boldsymbol{\Psi}'(\boldsymbol{z}_1)$, …, $\boldsymbol{\Psi}'(\boldsymbol{z}_m)$. This yields a fast emulator for y^M at any untried input $\boldsymbol{z}$,

$$\mathbb{E}(y^M(\boldsymbol{z}) \mid \boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M) = \boldsymbol{\Psi}'(\boldsymbol{z})\boldsymbol{\theta}^L + \boldsymbol{\gamma}_z'(\boldsymbol{\Gamma}^M)^{-1}(\boldsymbol{y}^M - \boldsymbol{X}\boldsymbol{\theta}^L)$$

$$\mathbb{V}\mathrm{ar}(y^M(\boldsymbol{z}) \mid \boldsymbol{y}^M, \boldsymbol{\theta}^L, \boldsymbol{\theta}^M) = \frac{1}{\lambda^M} - \boldsymbol{\gamma}_z'(\boldsymbol{\Gamma}^M)^{-1}\boldsymbol{\gamma}_z\,. \tag{1.1}$$

The variance in Equation (1.1) is zero for $\boldsymbol{z} \in D^M$. Therefore, the emulator given by *GaSP* is an interpolator of the data.

The *statistical structure* for implementing Step (5) must contend with model bias and uncertainty (Kennedy and O'Hagan, 2001), and thus is done as follows: View the computer model $y^M(\cdot)$ as a possibly biased representation of the un-

derlying real physical phenomenon $y^R(\cdot)$ by defining a bias process, b to satisfy $y^R(\cdot) = y^M(\cdot) + b(\cdot)$. Field data y^F are realizations of the real process

$$y^F(\cdot) = y^M(\cdot) + b(\cdot) + e(\cdot), \tag{1.2}$$

where $e(\cdot)$ is (field) measurement error. Arguments (inputs) of $y^R(\cdot), y^M(\cdot), b(\cdot), e(\cdot)$ will differ in kind depending on the specific situation being considered.

1.3 Outline of the Dissertation

This dissertation is concerned with validation of computer models with functional outputs, in the presence of uncertain inputs. We specifically consider the case where, given the input vector $\boldsymbol{z}$, the computer model produces a function of time, denoted as $y^M(\boldsymbol{z}, t)$. In practice, each functional output is given in terms of a vector, whose elements are evaluations of the function $y^M(\boldsymbol{z}, t)$ at grid points $t_1, \ldots, t_T$. We use $\boldsymbol{y}^M(\boldsymbol{z})$ to represent the output,

$$\boldsymbol{y}^M(\boldsymbol{z}) = \left(y^M(\boldsymbol{z}, t_1), \ldots, y^M(\boldsymbol{z}, t_T)\right)^t .$$

In addition, due to the fact that running the computer model is very time consuming, $\boldsymbol{y}^M(\boldsymbol{z})$ is only evaluated at selected values of $\boldsymbol{z}$, $\{\boldsymbol{z}_i, i = 1, \ldots, n\}$.

In many problems, $\boldsymbol{z}$ can be written as $\boldsymbol{z} = (\boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{u})$, where $\boldsymbol{v}$ is a vector of controllable inputs, $\boldsymbol{\delta}$ is a vector of unknown parameters that reflect key characteristics of the field runs, and $\boldsymbol{u}$ consists of calibration parameters. We use $\boldsymbol{\delta}_{ij}^*$ and $\boldsymbol{u}^*$ to represent the true values (related with the real process), for the j^{th} tested specimen in the i^{th} configuration $\boldsymbol{v}_i$, and the true calibration parameters, respectively. The statistical structure in Equation (1.2) suggests the following statistical model to connect the computer model with field runs,

$$y_r^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t) = y^M(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*; t) + b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t) + e_{ijr}(t)\,, \tag{1.3}$$

where y_r^F, y^M, b, e_r stand for the r^{th} field run, the computer model, the bias term, and the measurement error, respectively.

In Chapter 2, we consider the situation in which the responses are smooth functions of t, so that t can simply be treated as another input. The number of unknown $\boldsymbol{\delta}^*$ is as many as that of the field specimens, and must be dealt with in the analysis. We propose to incorporate the many $\boldsymbol{\delta}_{ij}^*$ by assuming a hierarchical structure for the biases. The approach will be discussed in the context of a problem involving a computer model of thermal flow in materials.

The major challenge in Chapter 3 and Chapter 4 is irregular functional time-data, precluding simply treating t as another input to the computer model. Indeed, to adequately represent an irregular function of time, so many time points would need to be included that the computation would become infeasible. The key components of the methodology consist of employing a basis representation over time and using *GaSP* to model the resulting basis coefficients. Chapter 3 uses a wavelet basis to represent the functional outputs. The study there only involves a single field specimen from a single computer code, which corresponds to the situation of a single $\boldsymbol{v}$, a single $\boldsymbol{\delta}^*$ and a single $\boldsymbol{u}^*$ in Equation (1.3).

Chapter 4 further extends this approach to model multiple computer codes with multiple field specimens for each $\boldsymbol{v}$. This involves multiple $\boldsymbol{\delta}^*$ and multiple $\boldsymbol{v}$ as in Chapter 2. In view of the fact that more computational effort is then needed, we reduce the number of coefficients to a manageable level by using eigen-basis elements, which are essentially principle components of the wavelet basis. Moreover, we build statistical surrogates to emulate computer model runs across all the codes and utilize a hierarchical structure of the biases due to the manufacturing variations. We will illustrate the analysis in Chapter 3 and Chapter 4 through a

test-bed computer model for analyzing stress on vehicle suspension systems that are subjected to forces over time.

Finally, in Chapter 5, we focus on computer models that produce time series data with an underlying stochastic structure. The grid points $t_1, \ldots, t_T$ are consecutive and equally spaced, labeled as $1, 2, \ldots, T$. Use of dynamic processes to emulate such models has the capability of capturing stochastic structure. We develop an approach that builds emulators using a multivariate Bayesian dynamic linear model. The resulting emulators also work as an interpolator of the data given the first few initial conditions.

Chapter 2

Dealing with Uncertain Inputs

The real physical process often involves uncertainty in some of the model inputs. We consider the two types of uncertain inputs that were introduced in Chapter 1, namely the calibration parameters $\boldsymbol{u}$ and the unknown parameters $\boldsymbol{\delta}$. In practice, $\boldsymbol{u}$ is often a vector of constants that are involved in the computer modeling. It does not vary over the different field runs. There is supposedly a true value for $\boldsymbol{u}$, denoted by $\boldsymbol{u}^*$, and the well-studied *calibration* problem is to learn about this parameter.

In contrast, less attention has been given to modeling possible uncertainty in $\boldsymbol{\delta}$, which can vary over the field runs. For instance, in many engineer applications, $\boldsymbol{\delta}$ is the manufacturing variation of the tested field specimen. In this chapter, we extend the *SAVE* methodology to incorporate uncertain inputs (both calibration and unknown), in the context of a fast computer model. We will illustrate the approach with a thermal computer model challenge problem (Dowding *et al.*, 2006).

2.1 Introduction and Statistical Model

Consider a computer model $y^M(\boldsymbol{z}, t)$, where $\boldsymbol{z}$ is the input and t is time. In this chapter, we assume that $y^M(\boldsymbol{z}, t)$ is fast, i.e. we can evaluate the computer code as many times as we wish.

Following the discussion in Chapter 1, we further write the input $\boldsymbol{z}$ as $(\boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{u})$, and denote by $(\boldsymbol{\delta}^*, \boldsymbol{u}^*)$ the true values in the real process. Letting $\boldsymbol{v}_i$ be the

controllable input for the i^{th} configuration and $\boldsymbol{\delta}_{ij}^*$ be the unknown input for the j^{th} tested specimen in the i^{th} configuration, we use the following statistical structure to connect the computer model with field runs,

$$y_r^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t) = y^M(\boldsymbol{v}_i, \boldsymbol{u}^*, \boldsymbol{\delta}_{ij}^*; t) + b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t) + e_{ijr}(t)\,, \tag{2.1}$$

where $y_r^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t)$, $b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t)$ and $e_{ijr}(t)$ stand for the r^{th} field replicate of the j^{th} tested specimen in the i^{th} configuration, the bias function associated with that specimen, and measurement errors, respectively. The $\boldsymbol{v}$ is a vector (of length p_v) which consists of the controllable inputs and $(\boldsymbol{\delta}, \boldsymbol{u})$ consists of unknown inputs. We assume that $\boldsymbol{u}^*$ is the same for all field runs, while $\boldsymbol{\delta}^*$ depends on the tested specimen in a particular field run. Some field runs may have the same values of $\boldsymbol{v}$ if they have the same configuration.

There are two challenges to analyzing the model in Equation (2.1).

Many unknowns. The $\boldsymbol{\delta}_{ij}^*$ are unknown and vary from specimen to specimen, leading to a large number of unknown parameters. When the $\boldsymbol{\delta}_{ij}^*$ arise as manufacturing variations from nominal values μ_i, a simple solution is to apply the traditional *SAVE* methodology by fixing $\boldsymbol{\delta}_{ij}^*$ at the nominal values. However, this ignores information inherent in the field data about the individual specimen-specific values. This can have a non-trivial effect on predictions and must be dealt with.

Structured bias. The bias term $b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t)$ in Equation (2.1) is associated with the particular individual tested specimen. Unfortunately, there is confounding between each $b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*; t)$ and the corresponding $\boldsymbol{\delta}_{ij}^*$, and, in general, no amount of data can sort this out. Moreover, there is very limited

data to estimate these different bias functions. We thus make the simplifying assumption that the bias depends only on the configuration $\boldsymbol{v}_i$. Understanding that this cannot be strictly correct, we then add in a different "nugget" error term for each replication to accommodate possible differences among the specimens, leading to the model

$$b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}; t) = b(\boldsymbol{v}_i; t) + \epsilon^b_{ij}(t)\,. \tag{2.2}$$

Thus the final model for analysis is

$$y^F_r(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*; t) = y^M(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*; t) + b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, t) + \epsilon^b_{ij}(t) + e_{ijr}(t)\,. \tag{2.3}$$

2.2 The Bayesian Analysis

The unknowns in the structure of Equation (2.3) are $\left(\boldsymbol{u}^*, \boldsymbol{\delta}^*_{ij}, b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, t), \epsilon^b_{ijr}(t)\right)$. The Bayesian analysis proceeds by placing prior distributions on the unknowns and then produces a posterior distribution of the unknowns given the data. The posterior distribution provides all the necessary information for prediction, tolerance bounds, etc.

2.2.1 The prior distributions

$\pi(\boldsymbol{u}^*)$ and $\pi(\boldsymbol{\delta}^*_{ij})$, the prior distributions for the uncertain inputs, is clearly context specific. For the unknown function $b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, t)$, we use a Gaussian process (GP) as its prior distribution. The GP is characterized by its mean and covariance function. As typical, we here take the mean of the GP to be an unknown constant μ_b. The covariance function of the GP is from a family whose parameters become part of the unknowns and are incorporated into the Bayesian analysis. Specifically,

$$b(\cdot,\cdot) \sim \mathrm{GP}\left(\mu_b, \tau^2 \mathbb{C}\left((\cdot,\cdot),(\cdot,\cdot)\right)\right),$$

where τ^2 is the variance of the GP, and the correlation function is assumed to be

$$\mathbb{C}\left(b(\boldsymbol{v};t), b(\boldsymbol{v}^{'};t^{'})\right) = c_v\left(\boldsymbol{v},\boldsymbol{v}^{'}\right) c_t\left(t,t^{'}\right), \tag{2.4}$$

with $c_v\left(\boldsymbol{v},\boldsymbol{v}^{'}\right) = \exp\left(-\sum_1^{p_v}\beta_k|v_k - v_k^{'}|^{\alpha_k}\right)$ and $c_t\left(t,t^{'}\right) = \exp\left(-\beta^{(t)}|t-t^{'}|^{\alpha^{(t)}}\right)$.

We model the $\epsilon_{ij}^b(t)$ and the $e_{ijr}(t)$ as independent Gaussian processes,

$$\epsilon_{ij}^b(t) \sim \text{GP}\left(0, \sigma_b^2 c_t(\cdot,\cdot)\right) \quad \text{and} \quad e_{ijr}(t) \sim \text{GP}\left(0, \sigma^2 c_t(\cdot,\cdot)\right).$$

In the above Gaussian processes, we adopt the correlation function $c_t(\cdot,\cdot)$ from Equation (2.4). This will dramatically simplify the computations as we shall see in Section 2.2.2.

Let $\boldsymbol{\theta}$ denote the hyper-parameters,

$$\boldsymbol{\theta} = \left(\sigma^2; \sigma_b^2; \tau^2; \alpha_1, \ldots, \alpha_{p_v}; \beta_1, \ldots, \beta_{p_v}; \alpha^{(t)}; \beta^{(t)}\right).$$

The prior $\pi(\boldsymbol{\theta})$ for these parameters is typically chosen in an objective fashion, but its complexity is often related to the nature and extent of the available data. Hence we defer further discussion of this choice until the applied example.

2.2.2 The posterior distribution

The likelihood function of the data, when combined with the prior distribution of the unknowns, leads to the posterior distribution, following Bayes Theorem. Let $(t_1, \ldots, t_T)$ be the time grid for the observations, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ be the configurations, n_i be the number of tested specimens with configuration $\boldsymbol{v}_i$, and r_{ij} be the number of replicated runs of the j^{th} tested specimen with configuration $\boldsymbol{v}_i$. We order the field data so that $y_k^F(t)$, the k^{th} field run is $y_r^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*; t)$, the r^{th} replicated run of the j^{th} tested specimen with configuration $\boldsymbol{v}_i$, $k = \sum_{s=1}^{i}\sum_{t=1}^{j-1} r_{st} + r$,

$i = 1, \ldots, m,\ j = 1, \ldots, n_i,\ r = 1, \ldots, r_{ij}$. Let $(\boldsymbol{v}_{(k)}, \boldsymbol{\delta}^*_{(k)})$ stand for the inputs associated with the k^{th} field run. The field runs with the same tested specimen have the same values for $\boldsymbol{v}_{(k)}$ and $\boldsymbol{\delta}^*_{(k)}$. Integrating out the bias function, we have

$$\boldsymbol{y}^F \sim \mathrm{N}\left(\boldsymbol{y}^M + \mu_b \mathbf{1}, \left(\tau^2 \boldsymbol{\Sigma}_b + \sigma_b^2 \boldsymbol{I} + \sigma^2 \boldsymbol{I}\right) \otimes \boldsymbol{\Sigma}_t\right), \tag{2.5}$$

where the vectors and matrices are defined as

$$\boldsymbol{y}^F = \left(y_k^F(t_l), k \in \{1, \ldots, \sum_{i=1}^{m} \sum_{j=1}^{n_i} r_{ij}\}, l \in \{1, \ldots, T\} \right)^t,$$

$$\boldsymbol{y}^M = \left(y^M(\boldsymbol{v}_{(k)}, \boldsymbol{\delta}^*_{(k)}, \boldsymbol{u}^*; t_l), k \in \{1, \ldots, \sum_{i=1}^{m} \sum_{j=1}^{n_i} r_{ij}\}, l \in \{1, \ldots, T\} \right)^t,$$

$$(\boldsymbol{\Sigma}_b)_{k,k'} = c_v(\boldsymbol{v}_{(k)}, \boldsymbol{v}_{(k')}), \qquad (\boldsymbol{\Sigma}_t)_{k,k'} = c_t(t_k, t_{k'}).$$

The Kronecker product operation $\otimes$ is defined in Appendix A and some of its properties are listed. These properties are crucial for easy evaluation of the likelihood.

To obtain the posterior distribution, we use what is called a modular MCMC approach, necessitated by the confounding between $(\boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)$ and the bias function. For explanation and justification, see Section 2.3. This begins by fixing $(\boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)$ at their prior means, and then running an MCMC for the other parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{b}, \mu_b, \tau^2, \sigma_b^2, \sigma^2)$. From this initial MCMC, we estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by their posterior medians $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$. Subsequently we view $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as fixed at these values, and rerun the MCMC with $(\boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)$ again viewed as unknown. The details are in Section 2.3.

Letting $\{\boldsymbol{b}\}$ be the vector of the bias function $b(\boldsymbol{v}; t)$ and given $(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$, the full

conditional distribution can be factorized as

$$\pi(\{\boldsymbol{b}\}, \{\boldsymbol{\delta}^*_{ij}\}, \boldsymbol{u}^*, \mu_b \mid \tau^2, \sigma_b^2, \sigma^2, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \boldsymbol{y}^F) = \pi(\{\boldsymbol{b}\} \mid \{\boldsymbol{\delta}^*_{ij}\}, \boldsymbol{u}^*, \mu_b, \tau^2, \sigma_b^2, \sigma^2, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \boldsymbol{y}^F) \times \pi(\mu_b, \{\boldsymbol{\delta}^*_{ij}\}, \boldsymbol{u}^* \mid \tau^2, \sigma_b^2, \sigma^2, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \boldsymbol{y}^F)\,. \tag{2.6}$$

The first factor in Equation (2.6) is a multivariate normal distribution, with mean vector and covariance matrix given by

$$\begin{aligned}
\mathbb{E}(\boldsymbol{b}) &= (\boldsymbol{y}^F - \boldsymbol{y}^M) - \tau^2 \boldsymbol{\Sigma}_b \left(\tau^2 \boldsymbol{\Sigma}_b + \sigma_b^2 \boldsymbol{I} + \sigma^2 \boldsymbol{I}\right)^{-1} \otimes \boldsymbol{I}(\boldsymbol{y}^F - \boldsymbol{y}^M - \mu_b \mathbf{1})\,, \\
\mathbb{C}(\boldsymbol{b}) &= \left((\sigma_b^2 + \sigma^2)(\sigma_b^2 \boldsymbol{I} + \sigma^2 \boldsymbol{I} + \tau^2 \boldsymbol{\Sigma}_b)^{-1} \tau^2 \boldsymbol{\Sigma}_b\right) \otimes \boldsymbol{\Sigma}_t\,.
\end{aligned} \tag{2.7}$$

2.3 MCMC Algorithm

To obtain draws from the second factor in Equation (2.6), we use a Gibbs sampler. The full conditional distribution for μ_b, $\pi(\mu_b \mid \{\boldsymbol{\delta}^*_{ij}\}, \boldsymbol{u}^*, \tau^2, \sigma_b^2, \sigma^2, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \text{Data})$, is normal with mean

$$\frac{\mathbf{1}^t \left(\boldsymbol{y}^F - \boldsymbol{y}^M\right)}{\mathbf{1}^t \left(\tau^2 \boldsymbol{\Sigma}_b + \sigma_b^2 \boldsymbol{I} + \sigma^2 \boldsymbol{I}\right)^{-1} \otimes \left(\boldsymbol{\Sigma}_t\right)^{-1} \mathbf{1}}\,,$$

and variance

$$\frac{1}{\mathbf{1}^t \left(\tau^2 \boldsymbol{\Sigma}_b + \sigma_b^2 \boldsymbol{I} + \sigma^2 \boldsymbol{I}\right)^{-1} \otimes \left(\boldsymbol{\Sigma}_t\right)^{-1} \mathbf{1}}\,.$$

The full conditional distribution for $\{\boldsymbol{\delta}^*_{ij}\}$ and $\boldsymbol{u}^*$, $\pi(\{\boldsymbol{\delta}^*_{ij}\}, \boldsymbol{u}^* \mid \mu_b, \tau^2, \sigma_b^2, \sigma^2, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}, \text{Data})$, does not have closed form. We use a Metropolis-Hastings algorithm to draw samples from this distribution. The proposal distribution depends on the context, and will be illustrated in the example.

Finally, conditional on $\{\boldsymbol{b}\}, \{\boldsymbol{\delta}^*_{ij}\}, \boldsymbol{u}^*, \mu_b$, $\widehat{\boldsymbol{\alpha}}$, $\widehat{\boldsymbol{\beta}}$, we draw samples from the posteriors of τ^2, σ_b^2 and σ^2. Their distributions will depend on the prior, and will be illustrated in the example.

We do the computation by the following modular MCMC approach, to improve the convergence of the Markov Chain. The algorithm has two separate MCMC steps - estimation of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and drawing samples of the remaining unknown quantities, given the estimate $\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}$. We will detail these two steps.

2.3.1 Estimation of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

Approximating the computer model output $y^M(\boldsymbol{v}, \boldsymbol{\delta}, \boldsymbol{u}; t)$ by $\widehat{y}^M = y^M(\boldsymbol{v}, \widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}; t)$, we can approximate the *SAVE* formula in Equation (2.3) as

$$y_r^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*; t) \approx \widehat{y}^M(\boldsymbol{v}_i, \widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}; t) + b(\boldsymbol{v}_i, t) + \epsilon_{ij}^b(t) + e_{ijr}\,, \tag{2.8}$$

where $(\widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}})$ is the vector of the prior means (nominal values) for $(\boldsymbol{\delta}, \boldsymbol{u})$. We use the following MCMC algorithm to draw from $\pi\left(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mu_b, \tau^2, \sigma_b^2, \sigma^2 \mid \boldsymbol{y}^F, \widehat{\boldsymbol{y}}^M\right)$.

At iteration h, $(h = 1, \ldots, 10000)$,

Step 1: Make draws of $\boldsymbol{\alpha}^h, \boldsymbol{\beta}^h$ conditional on $\mu_b^{(h-1)}$, $\tau^{2(h-1)}$, $\sigma_b^{2(h-1)}$, $\sigma^{2(h-1)}$, $\boldsymbol{y}^F$, $\widehat{\boldsymbol{y}}^M$. There are no closed forms for this distributions; we use the Metropolis-Hastings algorithm. The algorithm will be later detailed for the example.

Step 2: Given $\boldsymbol{\alpha}^h$, $\boldsymbol{\beta}^h$, $\tau^{2(h-1)}$, $\sigma_b^{2(h-1)}$, $\sigma^{2(h-1)}$, $\boldsymbol{y}^F$, $\widehat{\boldsymbol{y}}^M$, make a draw of μ_b^h according to the distribution in Section 2.2.2.

Step 3: Given $\boldsymbol{\alpha}^h$, $\boldsymbol{\beta}^h$, μ_b^h, $\tau^{2(h-1)}$, $\sigma_b^{2(h-1)}$, $\sigma^{2(h-1)}$, $\boldsymbol{y}^F$, $\widehat{\boldsymbol{y}}^M$, make draws of $\{\boldsymbol{b}^h\}$ according to Equation (2.7).

Step 4: Given $\boldsymbol{\alpha}^h$, $\boldsymbol{\beta}^h$, μ_b^h, and $\{\boldsymbol{b}^h\}$, make a draw of τ^{2h}, make a draw of σ_b^{2h}, and make a draw of σ^{2h}. Their distributions depend on the prior. We will give the distributions later for the example.

At the end of the MCMC, $\{\boldsymbol{\alpha}^h, \boldsymbol{\beta}^h, h = 1, \ldots, 10000\}$ are produced. We take the posterior medians of the MCMC draws as our estimates $\widehat{\boldsymbol{\alpha}}$, $\widehat{\boldsymbol{\beta}}$.

2.3.2 Modular MCMC algorithm

Given $\left(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}\right)$, the estimate we obtained above, the modular MCMC algorithm works as follows.

Step 0: Run the algorithm described above to obtain estimate for $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\beta}}$. We fix these parameters at their estimates for the rest of the analysis, and redo the MCMC, now incorporating the unknown $\{\boldsymbol{\delta}_{ij}^*\}$ and $\boldsymbol{u}^*$.

Step 1: At iteration h,

Step 1.1: Given $\mu_b^{(h-1)}$, $\tau^{2(h-1)}$, $\sigma_b^{2(h-1)}$, $\sigma^{2(h-1)}$, make draws for $\{\boldsymbol{\delta}_{ij}^{*h}\}, \boldsymbol{u}^*$ by Metropolis-Hastings algorithm. This depends on the context and we will give details for the example.

Step 1.2: Given $\{\boldsymbol{\delta}_{ij}^{*h}\}$, $\boldsymbol{u}^{*h}$, $\tau^{2(h-1)}$, $\sigma_b^{2(h-1)}$, $\sigma^{2(h-1)}$, make a draw for μ_b^h according to the distribution given in Section 2.2.2.

Step 1.3: Given $\{\boldsymbol{\delta}_{ij}^{*h}\}$, $\boldsymbol{u}^{*h}$, μ_b^h, $\sigma_b^{2(h-1)}$, $\sigma^{2(h-1)}$, $\tau^{2(h-1)}$, $\boldsymbol{y}^F$, $\widehat{\boldsymbol{y}}^M$, make draws of $\{\boldsymbol{b}^h\}$ according to Equation (2.7).

Step 1.4: Given μ_b^h, $\{\boldsymbol{\delta}_{ij}^{*h}\}$, $\boldsymbol{u}^*$ and $\{\boldsymbol{b}^h\}$, make a draw of τ^{2h}, a draw of σ_b^{2h}, and a draw of σ^{2h}. These distributions again depend on the prior, and will be given later for the example.

2.3.3 Extrapolation of the bias

We can extrapolate the bias to a new configuration $\boldsymbol{v}_{new}$. Denote the bias at the new configuration by $\boldsymbol{b}(\boldsymbol{v}_{new}) = (b(\boldsymbol{v}_{new}, t_k), k \in (1, \ldots, T))'$, and the biases for all the field data configurations by $\boldsymbol{b} = \{\boldsymbol{b}_i\}$. The posterior distribution for $\boldsymbol{b}(\boldsymbol{v})$ is

$$\pi(\boldsymbol{b}(\boldsymbol{v}) \mid \boldsymbol{y}^F) = \int \pi(\boldsymbol{b}(\boldsymbol{v}) \mid \boldsymbol{b}, \boldsymbol{\theta}) \pi(\boldsymbol{b}, \boldsymbol{\theta} \mid \boldsymbol{y}^F) \mathrm{d}\boldsymbol{b} \mathrm{d}\boldsymbol{\theta} \,.$$

We can make draws from this distribution as follows. At iteration h, in the MCMC described above, we have draws from $\boldsymbol{b}, \boldsymbol{\theta} \mid \boldsymbol{y}^F$. It then suffices to draw $\boldsymbol{b}^h(\boldsymbol{v})$ from $\pi\left(\boldsymbol{b}(\boldsymbol{v}) \mid \boldsymbol{b}^h, \boldsymbol{\theta}^{(h)}\right)$, which under the Gaussian process prior, is a normal distribution with mean and covariance given by

$$\mathbb{E}(\boldsymbol{b}(\boldsymbol{v})) = \mu_b \mathbf{1} + \boldsymbol{c}^t \left(\boldsymbol{\Sigma}_b \otimes \boldsymbol{\Sigma}_t\right)^- (\boldsymbol{b} - \mu_b \mathbf{1}), \quad \mathbb{C}\mathrm{ov}\left(\boldsymbol{b}(\boldsymbol{v})\right) = \tau^2 \left(1 - \boldsymbol{c}^t \boldsymbol{\Sigma}_b^{-1} \boldsymbol{c}\right) \boldsymbol{\Sigma}_t \,,$$

where $\boldsymbol{c} = (c_v(\boldsymbol{v}_1, \boldsymbol{v}), \ldots, c_v(\boldsymbol{v}_m, \boldsymbol{v}))^t$ is the correlation vector between the new configuration and the experimented configurations.

This modular MCMC approach results in a sequence of draws from the posterior distribution of all unknowns given the data. Statistical inference is based on these posterior draws.

We will devote the rest of this chapter to the analysis, feeding backward to improve the current validation scheme and computer model, and feeding forward to future validation activities. The approach will be illustrated with the example of the thermal computer model challenge problem (Dowding *et al.*, 2006).

2.4 An Example

2.4.1 The thermal computer model challenge problem

Dowding *et al.* (2006) proposes a 1-d thermal computer model as a challenge problem for validation. This thermal computer model simulates transient heat conduction through a slab. The output of the thermal computer model is

$$y^M(\kappa, \rho, T_0, x, L, q; t) = T_0 + \frac{qL}{\kappa}\left[\frac{\kappa t/\rho}{L^2} + \frac{1}{3} - \frac{x}{L} + \frac{x^2}{2L^2} - \sum_{N=1}^{6} \frac{2}{\pi^2 n^2} \exp\left(-\frac{n^2\pi^2\kappa t}{L^2\rho}\right) \cos\left(n\pi\frac{x}{L}\right)\right], \quad (2.9)$$

where κ is the thermal conductivity of the device, ρ is the volumetric heat capacity, q is applied heat flux, L = thickness, x = distance from the surface, T_0 = initial temperature and t is time. The inputs, (κ, ρ), are physical properties varying from specimen to specimen; they are unknown for a particular device. T_0 is fixed at 25^oC for all data and analyses and is therefore ignored. The controllable inputs (x, L, q) are assumed to be known exactly and their specification corresponds to a configuration.

In the general notation of Equation (2.3), the unknown input vector $\boldsymbol{\delta}$ consists of the two physical properties, $\boldsymbol{\delta} = (\kappa, \rho)$, that vary from specimen to specimen. The controllable inputs defining the configuration of the experiment are $\boldsymbol{v} = (x, L, q)$. The calibration parameter $\boldsymbol{u}$ does not exist in this example.

Let $y^R(\kappa, \rho, x, L, q; t)$ be the real temperature at time t for a specimen with properties κ, ρ under the associated experimental configuration. The principal application is to predict the (real) temperature at $t = 1000$ under the regulatory configuration $(x = 0, L = 0.019, q = 3500)$, and determine whether

$$P[y^R(\kappa, \rho, x = 0, L = 0.019, q = 3500; t = 1000) > 900] < .01, \quad (2.10)$$

the stated regulatory requirement. Because κ, ρ are unknown, interpretation of this probability must be dealt with. In fact, the Bayesian analysis we use treats these unknowns as random and their distribution is incorporated into the calculation of the probability.

There are three sets of field (experimental) data. The material characterization data (MC) are used to provide prior distributions for the κ, ρ's that go with each specimen. The ensemble data (EN) are used to produce assessments of the bias as well as tolerance bounds on model and reality predictions and are then further used to compare the predictive distribution $\pi\left(y^R\left(\kappa, \rho, x = 0, L = .019, q = 3000, t\right) \mid \text{EN}\right)$ with the accreditation configuration data (AC). The EN and AC data are then taken together and lead to a follow-on analysis providing new assessments of bias and tolerance bounds for predictions. This second analysis is used to predict temperature at the regulatory configuration.

Each of the EN and AC data has its own (unknown) κ, ρ and so there are as many parameters κ_i, ρ_i as there are EN and AC measurements. These many unknowns are assumed to have a common prior distribution. Additionally, the AC data are observed on a finer grid than the EN data. Instead of investing in extra computational effort we choose an innocuous simplification by only using the AC data on the common, albeit coarser, time grid. The purpose of this simplification is to achieve computational efficiency by utilizing a Kronecker product specification for the correlation matrices of the involved Gaussian processes.

The analyses are carried out for two situations: the so-termed medium-level data and the high-level data; the medium-level data is a subset of the high-level data. There are some limited data with $x \neq 0$ in the accreditation data set but we ignore them because only surface temperature ($x = 0$) is involved in the intended

application (regulatory condition) and little benefit is expected by including them.

We summarize the six steps of *SAVE* procedure for the thermal computer model challenge problem as follows.

(1) We give the I/U map for the thermal challenge problem in table 2.1.

Input	Impact	Uncertainty	Current status
κ	5	$\pi(\kappa)$	unknown
ρ	5	$\pi(\rho)$	unknown
q	5	None	1000, 2000, 3000
L	5	None	0.0127, 0.019, 0.0254
x	1	None	0
T_0	1	None	25
t	5	None	0, 50, 100, ..., 1000

Table 2.1: the Input/Uncertainty Map for thermal computer model. $\pi(\kappa), \pi(\rho)$ is given in section 2.4.2.

(2) We identify the evaluation criteria as the regulatory requirement in Equation (2.10).

(3) Field runs are available at the configurations defined in Table 2.1. The computer model runs are also available as in Equation (2.9).

(4) We bypass this step because the thermal computer model in Equation (2.9), is fast and can be evaluated as many times as we wish.

(5) The statistical structure to compare the computer model output with field data is given by Equation (2.3).

(6) We will discuss later in this chapter the feed backward and feed forward analysis.

2.4.2 Material Characterization

The MC data are used to obtain posterior distributions for κ and ρ that are then used as priors in later analyses. Figure 2.1 shows the quantile-quantile plots of the normalized data for all MC data. Though the plots suggest that κ and ρ might be assumed to be normally distributed, $\kappa \sim \mathrm{N}(\mu_\kappa, \sigma_\kappa^2)$ and $\rho \sim \mathrm{N}(\mu_\rho, \sigma_\rho^2)$, closer examination of the data for κ indicates that the assumption of constancy is not tenable; κ is, more plausibly, a linear function of temperature. But replacing the constant κ by a linear function in Equation (2.9) does not conform to the physics. Therefore, we only use the data at temperatures 500^oC or higher, to estimate $(\mu_\kappa, \sigma_\kappa^2, \mu_\rho, \sigma_\rho^2)$, in the hope that doing so will make $\pi(\kappa \mid MC)$ and $\pi(\rho \mid MC)$ close to the distributions under the regulatory condition.

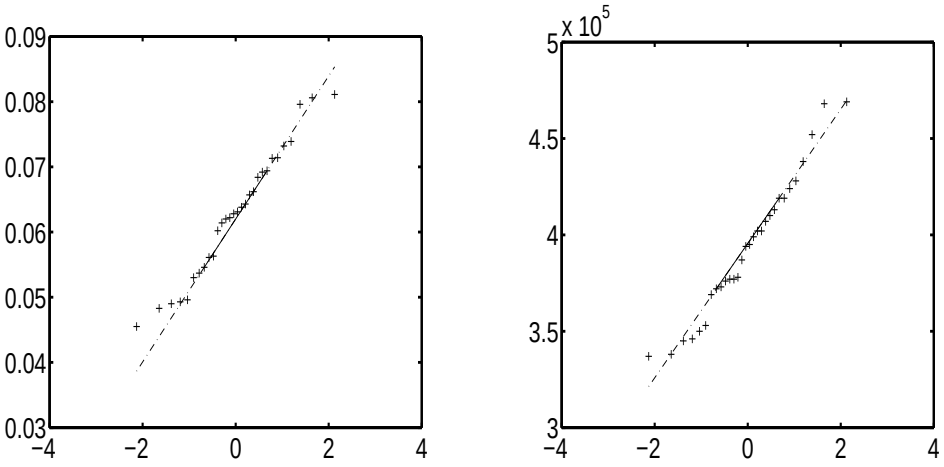

Figure 2.1: Left: qq-norm plot of κ; Right: qq-norm plot of ρ

The assumption that κ and ρ are independent is borne out by the data. The parameters of the normal distribution of κ are estimated in the traditional way as $\bar{\kappa}^{mc}, \sum_i (k_i^{mc} - \bar{\kappa}^{mc})^2/(n-1)$ and similarly for ρ. Table 2.2 gives the priors $\pi(\kappa)$ and $\pi(\rho)$ for medium and high level experimental data that result.

Prior	Medium	High
$\pi(\kappa)$	$\mathrm{N}(0.0671, 0.0070^2)$	$\mathrm{N}(0.0687, 0.0072^2)$
$\pi(\rho)$	$\mathrm{N}(405420, 38432^2)$	$\mathrm{N}(398220, 33690^2)$

Table 2.2: Prior distributions for κ and ρ

A naive (and generally wrong) answer to the question "Does the device meet the regulatory requirement?" is to sample κ and ρ from these priors, plug them, along with the regulatory configuration, into Equation (2.9), and count the proportion of times the regulatory criterion is violated. In Figure 2.2, we show the histograms (for medium- and high-level data) of the temperatures so obtained. The proportions above 900 are 0.08 and 0.06 for the two levels respectively. An initial conclusion is that the device might not be safe for the intended application. But we are not predicting the correct quantity: the correct quantity is reality at the regulatory configuration and bias, if present, must be accounted for before making conclusions.

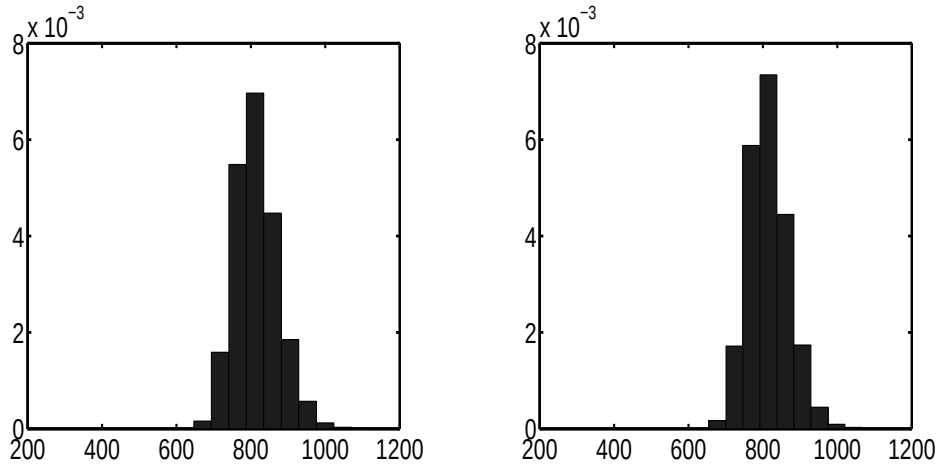

Figure 2.2: Computer model predictions for the surface temperature at regulatory configuration based on medium (left)- and high (right)- level MC data

2.4.3 Assumptions and Analysis

With x fixed at 0 the controllable inputs defining a configuration are $\boldsymbol{v} = (L, q)$. The prior distribution for the $\boldsymbol{\delta}^*_{ij}$ is given in Table 2.2. Now we specify the prior distributions for the hyper-parameters. Data limitations and the belief that the responses are smooth functions of input lead us to fix the α_i's at 2 (but not $\alpha^{(t)}$). We specify prior distributions for the other parameters as:

$$\begin{aligned}
&\pi(\sigma_b^2) \propto 1/\sigma_b^2, && \pi(\tau^2) \propto \exp(-1000/\tau^2),\\
&\pi(\beta_1) \propto \exp(-0.001\beta_1), && \pi(\beta_2) \propto \exp(-10^5\beta_2),\\
&\pi(\alpha^{(t)}) \propto I_{(1,2)}(\alpha^{(t)}), && \pi(\beta^{(t)}) \propto \exp(-100\beta^{(t)}) I_{[10^{-4},\infty)}(\beta^{(t)}),
\end{aligned}$$

where $I_A(x)$ equals one if $x \in A$ and zero otherwise, is the indicator function of the set A. We choose the standard objective prior for σ_b^2 because there are multiple field runs at each configuration. The remaining priors are chosen to be mildly informative due to the sparsity of the real data, and are chosen to reflect the scales of the parameters. For example, exploratory analysis suggests that β_1 is around 1000. Hence, we choose 0.001 as the rate parameter of its exponential prior. $\beta^{(t)}$ is truncated to guarantee non-singularity of Σ_t and avoid numerical issues. We fix σ^2 at 0 since the field data is assumed to be measured with no error and $r_{ij} = 1$.

These priors induce the posterior conditional distributions for τ^2 and σ_b^2 in Section 2.2.2 as follows.

$$\tau^2 \sim \text{IG}\left(\frac{n}{2}\text{rank}(\boldsymbol{\Sigma}_b) - 1, \frac{1}{2}(\boldsymbol{b} - \mu_b\mathbf{1})^t \left(\boldsymbol{\Sigma}_b \otimes \boldsymbol{\Sigma}_t\right)^{-1} (\boldsymbol{b} - \mu_b\mathbf{1}) + 1000\right), \quad (2.11)$$

and

$$\sigma_b^2 \sim \text{IG}\left(\frac{mn}{2}, \frac{1}{2}\left(\boldsymbol{y}^F - \boldsymbol{y}^M - \boldsymbol{b}\right)^t \left(\boldsymbol{I} \otimes \boldsymbol{\Sigma}_t\right)^{-1} \left(\boldsymbol{y}^F - \boldsymbol{y}^M - \boldsymbol{b}\right)\right). \quad (2.12)$$

In the MCMC algorithm, the step involving $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is as follows. We propose a new value of $\alpha^{(t)}$ by $\alpha^{(t)} = \alpha^{(t)h} + \mathrm{N}(0, 0.01^2)$, and then we calculate the acceptance ratio

$$\rho = \min \left(\frac{\mathrm{L}\left(\widehat{\boldsymbol{y}}^M, \mu_b^{(h-1)}, \tau^{2(h-1)}, \sigma_b^{2(h-1)}, \alpha^{(t)}, \boldsymbol{\beta}^{(h-1)}; \boldsymbol{y}^F\right)}{\mathrm{L}\left(\widehat{\boldsymbol{y}}^M, \mu_b^{(h-1)}, \tau^{2(h-1)}, \sigma_b^{2(h-1)}, \alpha^{(t)(h-1)}, \boldsymbol{\beta}^{(h-1)}; \boldsymbol{y}^F\right)} \, I_{(1,2)}(\alpha^{(t)}), 1 \right) ,$$

where L is the likelihood function, defined through Equation (2.5). Set $\alpha^{(t)h} = \alpha^{(t)}$ with probability ρ, and $\alpha^{(t)(h)} = \alpha^{(t)(h-1)}$ with probability $(1 - \rho)$. (Recall that α_i, $i = 1, 2$, are fixed throughout at 2.)

New values of $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta^{(t)})$ are proposed by $\beta_i \sim \mathrm{N}(\beta_i^{(h-1)}, s_i^2)$, $i = 1, 2$, and $\beta^{(t)} = \beta^{(t)(h-1)} + \mathrm{N}\left(0, \frac{1}{16}(\beta^{(t)(h-1)} - 10^{-4})^2\right)$, where $s_1 = 500$ and $s_2 = 10^{-7}$. The acceptance ratio is

$$\rho = \min \left(\frac{\mathrm{L}\left(\widehat{\boldsymbol{y}}^M, \mu_b^{(h-1)}, \tau^{2(h-1)}, \sigma_b^{2(h-1)}, \alpha^{(t)h}, \boldsymbol{\beta}; \boldsymbol{y}^F\right) \pi(\boldsymbol{\beta})}{\mathrm{L}\left(\widehat{\boldsymbol{y}}^M, \mu_b^{(h-1)}, \tau^{2(h-1)}, \sigma_b^{2(h-1)}, \alpha^{(t)h}, \boldsymbol{\beta}^{(h-1)}; \boldsymbol{y}^F\right) \pi(\boldsymbol{\beta}^{(h-1)})}, 1 \right) .$$

We set $\boldsymbol{\beta}^{(h)} = \boldsymbol{\beta}$ with probability ρ, and $\boldsymbol{\beta}^{(h)} = \boldsymbol{\beta}^{(h-1)}$ with probability $1 - \rho$. The Metropolis-Hastings algorithm usually yields highly correlated samples. We thus cycle through the algorithms 200 times saving only the last draws before proceeding to the next step of the MCMC.

The Metropolis-Hastings algorithm to draw samples for $\{\boldsymbol{u}_{ij}^*\}$ in Section 2.3 proceeds as follows. We propose new values by $\kappa_i = \kappa_i^{(h-1)} + \mathrm{N}(0, \frac{1}{20}\sigma_\kappa^2)$, $\rho_i = \rho_i^{(h-1)} + \mathrm{N}(0, \frac{1}{20}\sigma_\rho^2)$, and calculate the acceptance ratio as

$$\rho = \min \left\{ \frac{\mathrm{L}\left(\boldsymbol{y}^M(\boldsymbol{u}_{ij}^*), \mu_b^{(h-1)}, \tau^{2(h-1)}, \sigma_b^{2(h-1)}; \mathrm{Data}\right) \prod_i \pi(\boldsymbol{u}_{ij}^*)}{\mathrm{L}\left(\boldsymbol{y}^M(\boldsymbol{u}_{ij}^{*(h-1)}), \mu_b^{(h-1)}, \tau^{2(h-1)}, \sigma_b^{2(h-1)}; \mathrm{Data}\right) \prod_i \pi(\boldsymbol{u}_{ij}^{*(h-1)})}, 1 \right\} .$$

We set $\{\boldsymbol{u}_{ij}^{*h}\} = \{\boldsymbol{u}_{ij}^{*}\}$ with probability ρ, and $\{\boldsymbol{u}_{ij}^{*(h-1)}\}$ otherwise. Again, we cycle through 200 times at each iteration, and save only the last draws of the parameters.

2.4.4 Results

The MCMC produces a sequence of draws $\left(\{\kappa_i^h, \rho_i^h\}, \mu_b^h, \tau^{2h}, \sigma_b^{2h}\right)$. $\boldsymbol{y}^{Mh}$ is obtained by plugging the draws $\left(\kappa_i^h, \rho_i^h\right)$ into Equation (2.9) and evaluating. Also obtain $\boldsymbol{b}^h$ by drawing from multivariate normal distribution in Equation (2.7).

We obtain 10000 such draws. With these MCMC draws, we obtain the model prediction (sometimes called the pure model prediction) by averaging $\boldsymbol{y}^{Mh}$ over h. Call the result $\widehat{\boldsymbol{y}}^M$. Because the (κ_i, ρ_i)'s are different for each replicate, the predictions will differ from replicate to replicate. Reality at a specimen in the experiment is the same as the field value, since there is no measurement error. It follows that a 95% point-wise tolerance bound at time t for the model prediction at such a specimen can be obtained as the set of values $\delta^M(t)$, such that 95% of the $y^{Mh}(t)$ for this specimen satisfy $|y^{Mh}(t) - \hat{y}^M(t)| < \delta^M(t)$ for all t.

For predicting at a new configuration, $\boldsymbol{v}_{new}$ (and therefore a new specimen with parameters κ_{new}, ρ_{new}), we first generate $\boldsymbol{b}^h(\boldsymbol{v}_{new}) = \left(b^h(\boldsymbol{v}_{new}, t_j), j = (1, \ldots, n)\right)'$ by drawing from the multivariate normal generated by the GP assumption on b while conditioning on the data and the draws on all parameters including $\boldsymbol{b}^h$. Then add $\boldsymbol{\epsilon}^h$ to $\boldsymbol{b}^h(\boldsymbol{v}_{new})$, where $\boldsymbol{\epsilon}^h \sim \mathrm{N}(\mu_b \mathbf{1}, \sigma_b^{2h} \widehat{\boldsymbol{\Sigma}}_t)$. Then generate $\boldsymbol{y}^{Mh}(\kappa_{new}^h, \rho_{new}^h, \boldsymbol{v}_{new})$ by drawing $\kappa_{new}^h, \rho_{new}^h$ from their prior distributions and plugging them into Equation (2.9). Draws of reality, $\boldsymbol{y}^R = \boldsymbol{y}^M(\kappa_{new}, \rho_{new}, \boldsymbol{v}_{new}) + \boldsymbol{b}(\boldsymbol{v}_{new}) + \boldsymbol{\epsilon}$, can also be obtained. Letting $\boldsymbol{y}^{Rh}$ be the MCMC draws of $\boldsymbol{y}^R$, we have reality prediction, $\widehat{\boldsymbol{y}}^R$, as the average of the $\boldsymbol{y}^{Rh}$ and a 95% point-wise tolerance bound can be obtained

as the $\delta^R(t)$ such that 95% of the h satisfy $|y^{Rh}(t) - \widehat{y}^R(t)| < \delta^R(t)$ for all t.

We next give the results conditional on EN data, conditional on EN+AC data, and the regularity assessment.

Ensemble analysis

For the EN data, the posterior distributions of the parameters $\boldsymbol{\theta} = (\mu_b, \boldsymbol{\alpha}, \boldsymbol{\beta}, \tau^2, \sigma_b^2)$ based on the analysis described above are given in Table 2.3. (α_1, α_2) are fixed at 2 due to the limitation of the data, reflecting our belief of the smoothness of the outputs as functions of L and q. In Figure 2.3, we see the scatter plots of the

Parameter	Medium Level	High Level
β_1	$549.96(47.67, 2314.27)$	$851.41(227.30, 2660.50)$
β_2	$1.80(0.37, 6.21) \times 10^{-7}$	$3.25(1.23, 8.91) \times 10^{-7}$
$\alpha^{(t)}$	$1.9987(1.9969, 1.9995)$	$1.9989(1.9980, 1.9994)$
$\beta^{(t)}$	$1.33(1.09, 1.63) \times 10^{-3}$	$1.02(0.89, 1.15) \times 10^{-3}$
μ_b	$-0.57(-23.10, 23.38)$	$-11.56(-41.52, 18.15)$
τ^2	$269.72(148.86, 512.02)$	$473.89(251.25, 906.29)$
σ_b^2	$22.59(15.10, 35.53)$	$66.02(50.97, 87.30)$

Table 2.3: Posterior medians with 95% credible intervals for the indicated parameters given EN data

last 1000 $\{(\kappa_i^h, \rho_i^h)\}$ draws given the EN high level data. The four different colors correspond to the four replicates. The model prediction and the bias function for each of the replicates and each configuration can be calculated as described at the beginning of this Section. To illustrate, the upper left panel of Figure 2.4 shows the model prediction (solid black) with 95% tolerance bounds for the first replicate of the high level EN data at configuration $L = 0.0127, q = 1000$. The red curve plots the experimental data. In the upper right panel we have the bias plotted along with 95% uncertainty bounds for the same setting. For the same configuration and a new specimen, with parameters $(\kappa_{new}, \rho_{new})$, the lower right

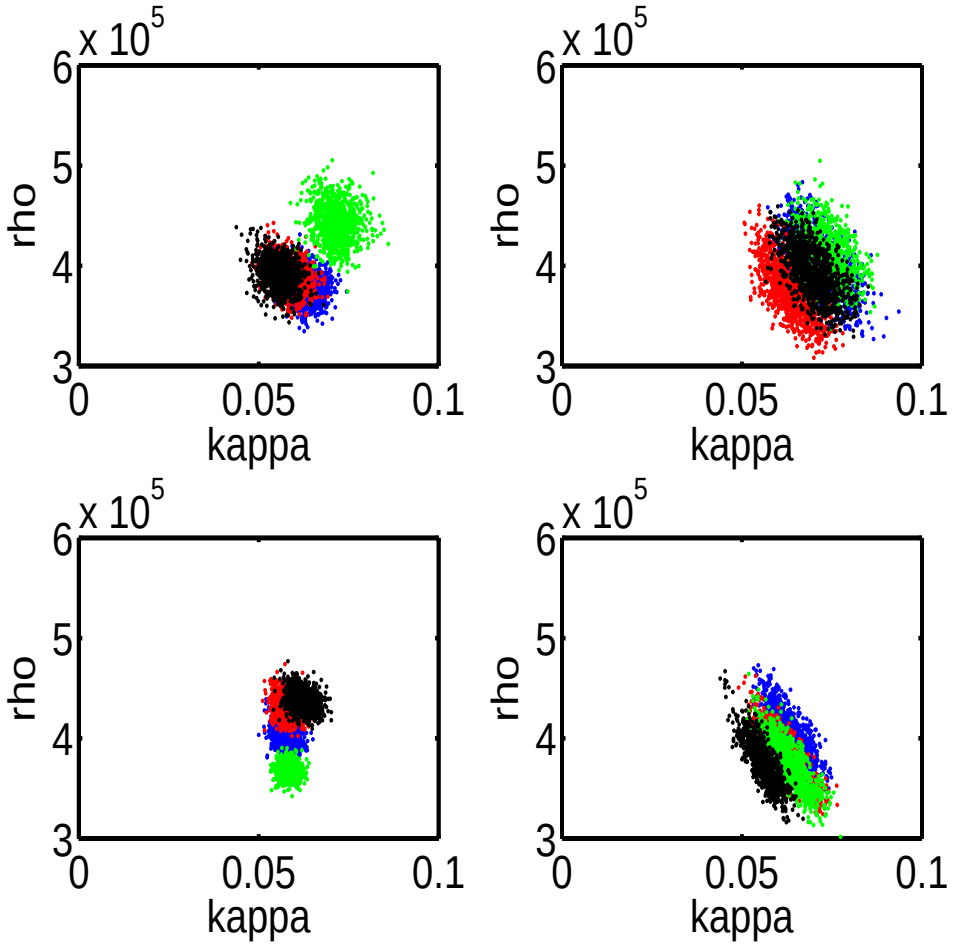

Figure 2.3: Scatter plots of (κ_i, ρ_i) given EN data (high level) with the configurations: $L = 0.0127, q = 1000$ (Upper-Left), $L = 0.0127, q = 2000$ (Upper-Right), $L = 0.0254, q = 1000$ (Lower-Left), $L = 0.0254, q = 2000$ (Lower-Right).

panel gives the corresponding result for the bias and the lower left panel gives the reality prediction (not the model prediction) as solid black line with dashed lines as tolerance bounds. For predicting (extrapolating) at the AC configuration $x = 0, L = 0.019, q = 3000$, a new pair $(\kappa_{new}, \rho_{new})$ is also involved. We can use the same $\boldsymbol{\theta}^h$'s found above but draw κ^h, ρ^h from their prior distribution. In addition, we must draw $\boldsymbol{\epsilon}$ from the distribution and draw from the distributions of $\boldsymbol{b}(0, 0.019, 3000)$ given the four values of $\boldsymbol{b}^h$ at the EN configuration. Every-

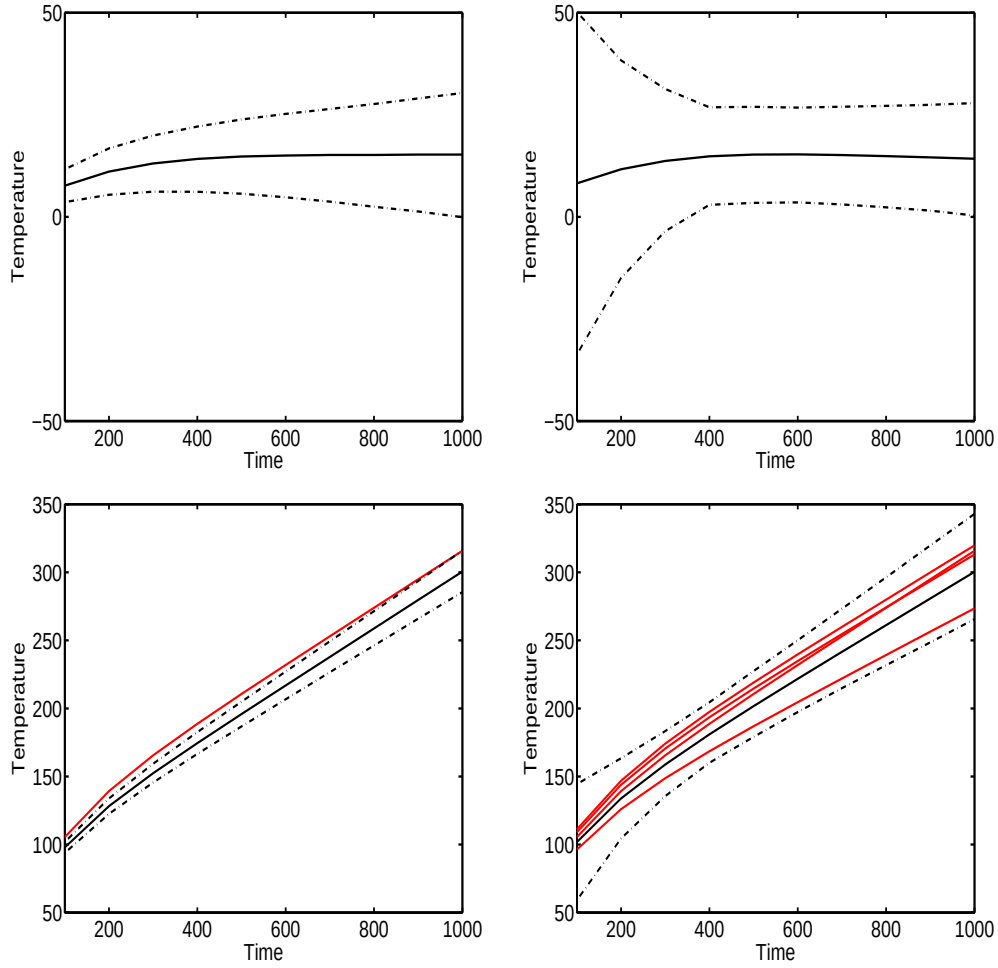

Figure 2.4: Bias function (upper-left) for the first run in the first ensemble configuration ($L = 0.0127, q = 1000$); bias function (upper-right) for a new specimen at this configuration; Model prediction (lower-left) for the first run of this configuration; and reality prediction (lower-right) for a new specimen at this configuration. The observations are plotted as red lines, the posterior medians as solid black lines, and the 2.5% and 97.5% point-wise confidence bands as dashed black lines.

thing else is done as above and produces the bias function in Figure 2.5, the model prediction in the left panel of Figure 2.6, and reality prediction for the AC configuration in the right panel.

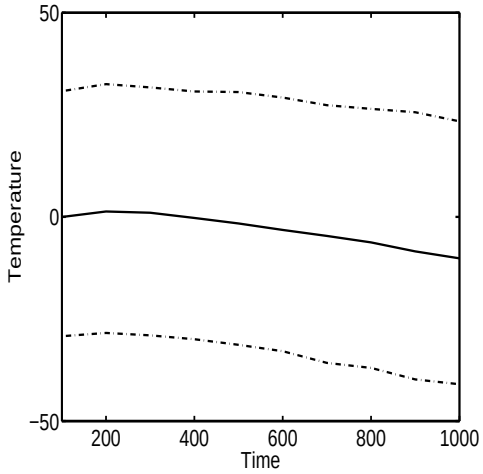

Figure 2.5: The bias function at the accreditation configuration $(L = 0.019, q = 3000)$ given high-level EN data.

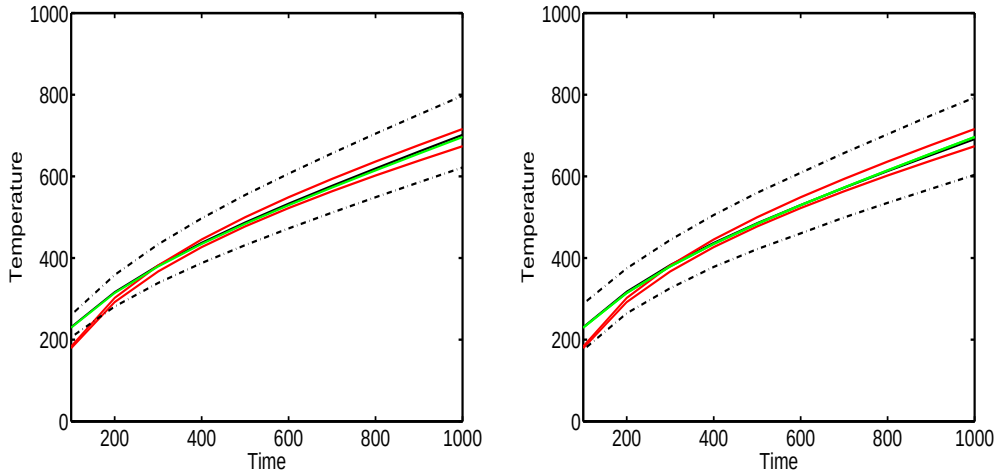

Figure 2.6: Pure model prediction (left) and Reality prediction (right) at the accreditation configuration with tolerance bounds in black; experimental data are in red; the green line is the prediction by plugging in the prior means of κ and ρ.

Accreditation analysis

With the addition of the accreditation data, a reprise of the ensemble analysis is summarized as follows:

- The posterior distribution of unknown parameters is shown in Table 2.4.

- Scatter plots of the last 1000 $\{(\kappa_i, \rho_i)\}$ draws for the two replicates (distinguished by color) in the AC configuration given high level EN + AC data are plotted in Figure 2.7.

- Figure 2.8, the counterpart to Figure 2.4, displays model prediction and bias prediction for the first AC replicate as well as the reality prediction and bias prediction for a new specimen with parameters κ_{new}, ρ_{new}.

Parameter	Medium Level	High Level
β_1	$11.93(0.92, 45.52)$	$17.42(4.22, 53.20)$
β_2	$1.30(0.62, 3.68) \times 10^{-6}$	$1.19(0.59, 3.43) \times 10^{-6}$
$\alpha^{(t)}$	$1.9967(1.9918, 1.9988)$	$1.9983(1.9970, 1.9989)$
$\beta^{(t)}$	$1.33(1.09, 1.71) \times 10^{-3}$	$1.05(0.94, 1.17) \times 10^{-3}$
μ_b	$-62.92(-140.65, 18.56)$	$-95.25(-245.31, 52.60)$
τ^2	$6500.37(4121.96, 10786.89)$	$22423.07(14239.77, 36243.67)$
σ_b^2	$9.84(6.50, 15.96)$	$43.97(34.12, 57.82)$

Table 2.4: Posterior medians with 95% credible intervals for the indicated parameters given EN + AC data.

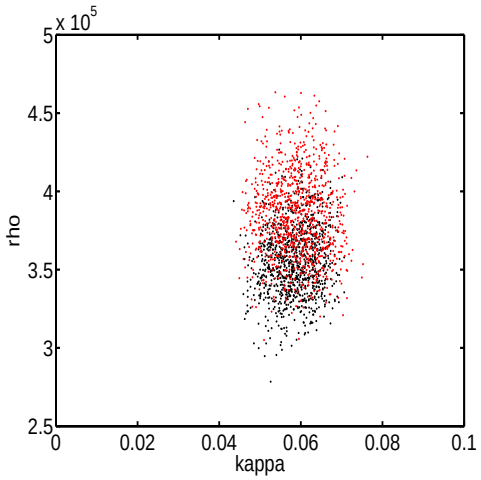

Figure 2.7: Scatter plots of (κ_i, ρ_i) for the 2 AC replicates given EN + AC data (high level)

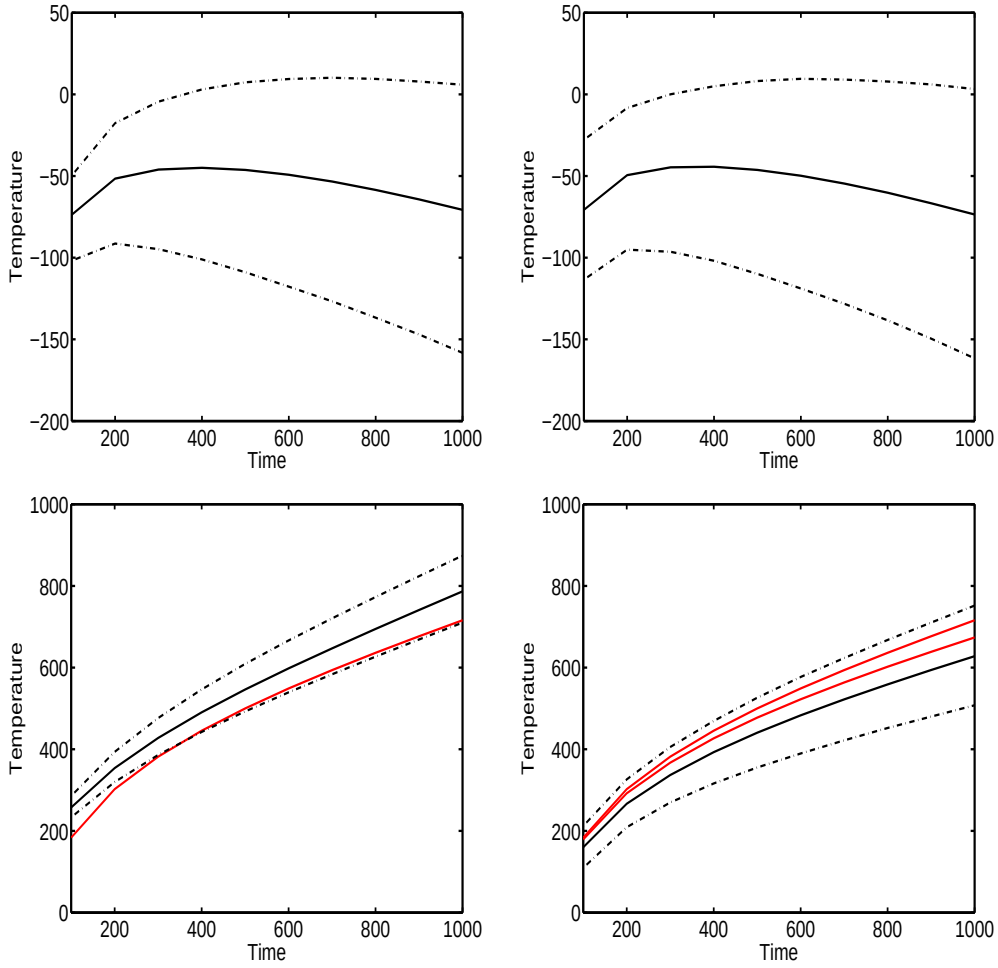

Figure 2.8: The bias function (upper-left) for the first run in the accreditation configuration ($L = 0.019, q = 3000$); bias function (upper-right) for a new run at this configuration; pure model prediction (lower-left) for the first run in this configuration; and reality prediction (lower-right) for a new run at this configuration. Red lines are the experimental data.

We first compare the results given EN data in Table 2.3 and the results given EN+AC data in Table 2.4. The GP parameters associated with time change very little; however β_1, β_2 are changed a lot, since the accreditation data adds a new configuration in the sparse design space of (L, q). The big change lies in the variances for the stochastic process and for the nugget. The variance in the

accreditation analysis is much larger than the variance in the ensemble analysis.

Next, we compare the bias function given EN data at accreditation configuration in Figure 2.5 with the bias function given EN+AC data in Figure 2.8. These are very different, since the accreditation data provides more information about the bias function at the accreditation configuration.

The pure model prediction in Figure 2.8 is also different from the one given in Figure 2.6. This is because the pure model prediction in Figure 2.6 only takes into account the variability of the input variables (κ, ρ), while the pure model prediction in Figure 2.8 has to take into account the bias structure defined in Equation (2.2).

The prediction of reality in Figure 2.8 also exhibits more variability than the prediction of reality in Figure 2.6.

The regulatory assessment

Let y_R^M, b_R, y_R^R be the model prediction, bias function and reality prediction, respectively, under the regulatory configuration $x_R = 0$, $L_R = 0.019$, $q_R = 3500$ at time $t_R = 1000$. We get posterior draws for y_R^M, b_R, y_R^R as in the discussion of bias extrapolation.

Figure 2.9 gives the posterior histograms of the reality prediction of the device surface temperature under the regulatory configuration at time 1000, given the EN+AC data at both medium and high levels. The distributions are summarized in Table 2.5.

The proportion of values that exceed 900 is the estimate of the probability that the regulatory requirement is unmet. For the medium level this number is 0.02; for high-level data the number is 0.04. Though chance of failure decreases,

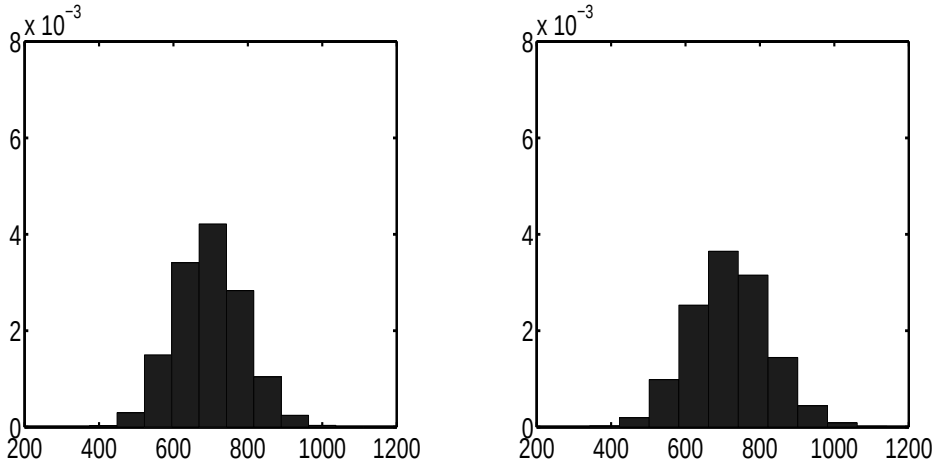

Figure 2.9: Histograms of the device surface temperature under regulatory configuration with medium (left)- and high (right)- level AC + EN data.

Value	Medium Level	High Level
Mean	697.13	719.20
Median	695.93	717.90
Standard deviation	91.52	105.32

Table 2.5: Summary of the device surface temperature under regulatory configuration

compared with the pure model predictions we have discussed in Section 2.4.2, the requirement of 0.01 is still not met.

2.5 Discussion

The formulation of the problem and the process described above provides an answer to the question of how to assess a computer model, in the presence of uncertain inputs. For computer models that are not fully reliable, such as the thermal problem, the ability to use the model is enhanced by producing "legitimate" estimates of reality (the reality predictions described above). The approach requires good prior knowledge of the unknown inputs $\boldsymbol{\delta}_{ij}^*$ and $\boldsymbol{u}^*$.

The type of computer model discussed in this chapter is special, in that we can evaluate the computer model as many times as we wish. Thus we bypass Step (4) in the *SAVE* procedure. In many applications, however, the computer models are computationally expensive and thus can be evaluated only at a limited number of input parameters. Under such circumstances, as we shall see in Chapter 4, *GaSP* approximation of the computer model can be combined with the hierarchical bias structure discussed in this chapter, still allowing the statistical inference in the presence of uncertain inputs for expensive-to-run computer models.

Chapter 3

Functional Output

This chapter is motivated by methodological needs in analyzing computer models that produce functional data. An example is given by a computer model that analyzes stress on a vehicle suspension system, subject to the forces over time (the problem is described in Section 3.1). We use this problem as our test bed example to illustrate the approach. Methodological development of the approach to deal with functional data, especially irregular functional data, greatly widens the applicability of Bayarri *et al.* (2005a)'s strategy.

The following important and technically challenging problems are involved and must be taken into account. First, the functions are not smooth in many engineering scenarios with functional output. Consequently, the approach in Bayarri *et al.* (2005b) can result in a computationally intractable problem. This is so in the test bed problem, for instance, with typical irregular functional data indicated in Figure3.1. A second ubiquitous problem in engineering scenarios is that (unmeasured) manufacturing variations are present in tested components; incorporating this uncertainty into the analysis, as in Chapter 2, can be crucial. Finally, the point of computer modeling in engineering contexts is typically to allow use of the computer model to predict outcomes in altered or new settings, for which no field data are available. We consider several approaches to this problem.

3.1 Introduction

The computer model we consider in this chapter is a time-dependent system. Given a vector of inputs, $\boldsymbol{x} = (\boldsymbol{v}, \boldsymbol{\delta})$, to the system, denote the "real" response over time t as $y^R(\boldsymbol{x}; t)$. Field measurement of the real response has error and we write the r^{th} replicate field measurement as

$$y_r^F(\boldsymbol{x}; t) = y^R(\boldsymbol{x}; t) + \varepsilon_r(t)\,, \tag{3.1}$$

where the $\varepsilon_r(\cdot)$'s are independent mean zero Gaussian processes that will be further discussed in Section 3.2.2. Some inputs may have error; we must also take that into account.

In addition, there is a computer model aimed at producing the same response. The computer model may have within it calibration/tuning parameters $\boldsymbol{u} = (u_1, \ldots, u_m)$ that need to be estimated (or tuned) to produce a matching response. The model output is then of the form $y^M(\boldsymbol{x}, \boldsymbol{u}; t)$; it is affected by $\boldsymbol{u}$ but the real response, y^R, is not. The connection between model output and reality is then expressed in

$$y^R(\boldsymbol{x}; t) = y^M(\boldsymbol{x}, \boldsymbol{u}^*; t) + b(\boldsymbol{x}; t)\,, \tag{3.2}$$

where $\boldsymbol{u}^*$ is the true value of the (vector) calibration parameter; $y^M(\boldsymbol{x}, \boldsymbol{u}^*; t)$ is the model response at time t and the true value of $\boldsymbol{u}$; and $b(\boldsymbol{x}; t)$, defined by subtraction, is the associated bias. In situations where $\boldsymbol{u}$ is a tuning parameter there is no "true value" so $\boldsymbol{u}^*$ should be thought of as some type of fitted value of $\boldsymbol{u}$, with the bias defined relative to it.

These two equations, (3.1) and (3.2) describe the calibration/validation structure we address. Data from the field and from the computer model runs provide information for estimating the unknowns in equation (3.1) and (3.2). The

Bayesian analysis we employ takes note of the fact, as in Bayarri *et al.* (2005a), that the unknowns $\boldsymbol{u}^*$ and the bias are not statistically identifiable and, consequently, specification of their prior distributions is of particular importance for the analysis.

An example computer model of this type is the test bed case study described as follows. The problem is to predict loads resulting from stressful events on a vehicle suspension system over time e.g., hitting a pothole. In the initial part of the study there are seven unmeasured parameters of the system with specified nominal (mean) values $\boldsymbol{x}_{nom}$ (referred to later on as Condition A), which corresponds to the configuration input $\boldsymbol{v}$ in Equation (1.3). Additionally, these parameters are subjected to unknown manufacturing variations, represented by $\boldsymbol{\delta}^*$. There are other relevant parameters that are known and fixed and hence not part of the experiments.

Field data are obtained by driving a vehicle over a proving ground course and recording the time history of load at sites on the suspension system. The curves must be registered (Appendix B) to assure that peaks and valleys occur at the same place.

In addition, there is a computer model aimed at producing the same response. The computer model is a so-termed ADAMS model, a commercially available, widely used finite-element based code that analyzes complex dynamic behavior (e.g., vibration, stress) of mechanical assemblies. The computer model has within it two calibration parametters $\boldsymbol{u}^* = (u_1^*, u_2^*)$ quantifying two different types of damping (unknown levels of energy dissipation) that need to be estimated (or tuned) to produce a matching response.

For proprietary reasons the specific parameters are not fully described – they

include characteristics of tires, bushings and bumpers as well as vehicle mass. In addition, the values assigned to these parameters are coded on a [0,1] scale and the ouput responses are also coded. In the coded scale the fixed values of $\boldsymbol{x}_{nom}$ are all 0.50. The uncertainty ranges for the nine parameters were elicited through extensive discussion with engineers and modelers; they are given in Table 3.1, the so-termed Input/Uncertainty map (Bayarri *et al.*, 2005a). Along with the ranges, prior distributions were elicited for $(\boldsymbol{u}^*, \boldsymbol{\delta}^*)$ in Section 3.2.4.

Parameter	**Type**	**Uncertainty Range**
$Damping_1$	Calibration	[0.125, 0.875]
$Damping_2$	Calibration	[0.125, 0.875]
x_1	Nominal+Variation	[0.1667, 0.8333]
x_2	Nominal+Variation	[0.1667, 0.8333]
x_3	Nominal+Variation	[0.2083, 0.7917]
x_4	Nominal+Variation	[0.1923, 0.8077]
x_5	Nominal+Variation	[0.3529, 0.6471]
x_6	Nominal+Variation	[0.1471, 0.8529]
x_7	Nominal+Variation	[0.1923, 0.8077]

Table 3.1: I/U Map. Uncertainty ranges for calibration parameters and parameters subject to manufacturing variation.

Field data:

In the initial study with Condition A inputs a single vehicle was driven over a proving ground course seven times. The recorded field data consist of the time history of load at two sites on the suspension system. Plots of the output for Site 1 can be seen in Figure 3.1 for two of the time periods of particular interest. Thus there are seven replicates and a single $\boldsymbol{x}_{nom}$ in the field data.

Computer model runs:

A typical model run for the test bed example takes one hour, limiting the number of runs that can feasibly be made. To select which runs to make we

adopted the design strategy used in Bayarri *et al.* (2005a):

The 9-dimensional rectangle defined by the ranges of the parameters in Table 3.1 is first transformed into the 9-dimensional unit cube. We then selected a 65 point Latin Hypercube Design (LHD) using code by W. Welch that finds an approximately maximin LHD. In addition, we added a point at the center (0.5,...,0.5), the nominal values. One run failed to converge and was deleted from the experiment leaving a total of 65 design points.

3.2 Formulation, Statistical Model and Assumptions

3.2.1 Formulation

We formulate the problem following the notation in Section 3.1. In addition, some inputs in Equation (3.2) may be specified or physically measured with essentially perfect accuracy. Those that remain fixed for both field data and model runs play no further role and are not part of $\boldsymbol{x}$. Other (unmeasured) inputs will have specified nominal values (generally, they will vary in the experiments) that are subject to manufacturing variation with specified distributions. We write these as

$$\boldsymbol{x} = \boldsymbol{x}_{nom} + \boldsymbol{\delta} \tag{3.3}$$

where $\boldsymbol{x}_{nom}$ is the known nominal value and the distribution of the manufacturing variation $\boldsymbol{\delta}$ can be specified. In effect this transforms Equation (3.1) and Equation (3.2) into

$$y^R(\boldsymbol{x}_{nom} + \boldsymbol{\delta}^*; t) = y^M(\boldsymbol{x}_{nom} + \boldsymbol{\delta}^*, \boldsymbol{u}^*; t) + b(\boldsymbol{x}_{nom} + \boldsymbol{\delta}^*; t) \tag{3.4}$$

and

$$y_r^F(\boldsymbol{x}_{nom} + \boldsymbol{\delta}^*; t) = y^R(\boldsymbol{x}_{nom} + \boldsymbol{\delta}^*; t) + \varepsilon_r(t)\,, \tag{3.5}$$

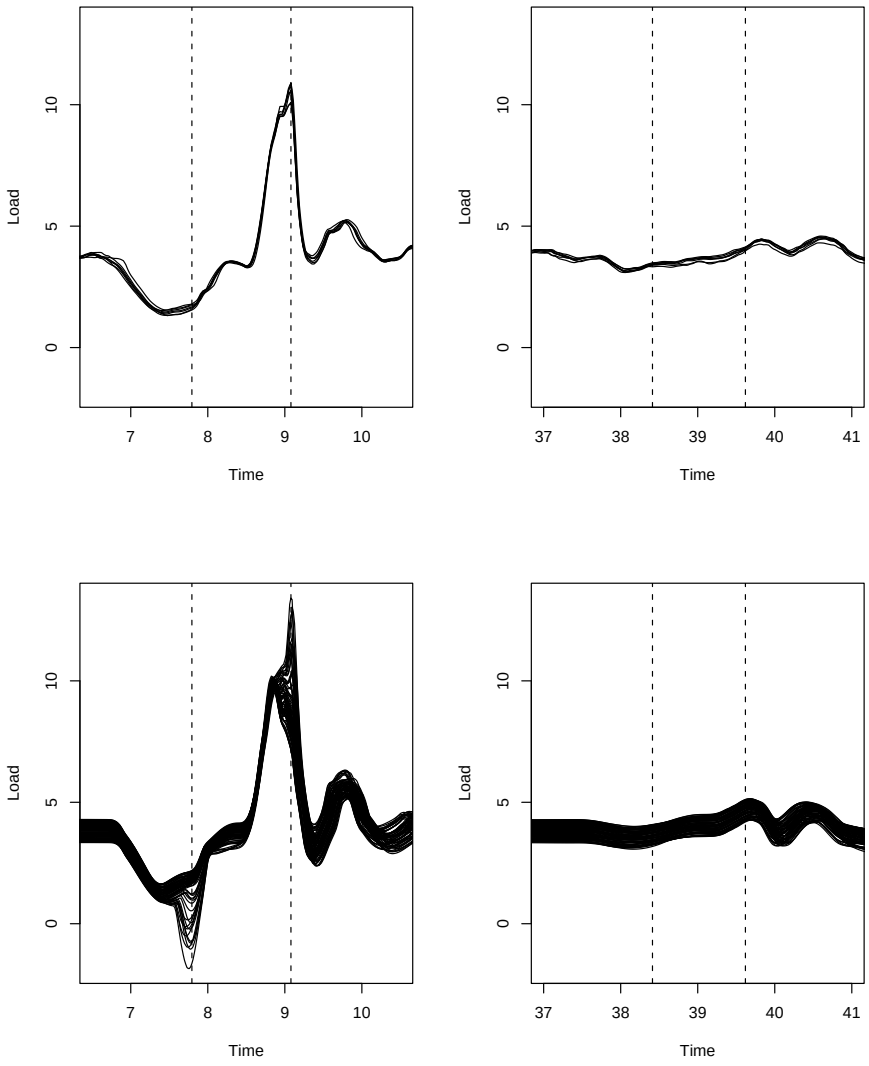

Figure 3.1: Model output (bottom) and registered field output (top) for Site 1 at Region 1 (left) and Region 2 (right). Vertical lines indicate the reference peak locations.

where $\boldsymbol{\delta}^*$ is the actual (unknown) value of $\boldsymbol{\delta}$. The parameters $\boldsymbol{\delta}$ are like calibration parameters in that they are unknown but physically real.

Prior to making computer runs or collecting field data, the unknowns in Equation (3.4) and Equation (3.5) are $\left(y^M, \boldsymbol{u}^*, \boldsymbol{\delta}^*, b, V_\epsilon\right)$, where V_ϵ is the covariance function of ϵ. A full Bayesian analysis would contemplate placing priors on these unknowns and, given field data and model runs, produce posterior distributions. But the complexity (for example, of irregular functional output) and high-dimensionality militate against such a strategy unless simplifications can be made. One such is the use of a basis representation of the functional data. In particular, to handle the irregular functions, we will consider wavelet decompositions. Other settings may allow different representations such as Fourier series or principal components (Higdon *et al.*, 2007).

3.2.2 Wavelet decomposition

The nature of the functions in Figure 1, for example, suggests that wavelet decomposition would be a suitable basis representation (see Vidakovic (1999);Müller and Vidakovic (1999) and Morris *et al.* (2003) are among the other references with applications related to ours).

The wavelet decomposition (more details are in Appendix B) we use for y^M is of the form

$$y^M(\boldsymbol{x}, \boldsymbol{u}; t) = \sum_i w_i^M(\boldsymbol{x}, \boldsymbol{u})\Psi_i(t) \tag{3.6}$$

where the wavelet basis functions $\Psi_i(t)$ are default choices in *R wavethresh* (Daubechies wavelets of index 2; for simplicity of notation, we include the scaling function as one of the basis elements). Similarly, the field curves (rth replicate) are repre-

sented as

$$y_r^F(\boldsymbol{x};t) = \sum_i w_{ir}^F(\boldsymbol{x})\Psi_i(t)\,. \tag{3.7}$$

A thresholding procedure, used to produce a manageable number of coefficients while maintaining adequate accuracy, leads to the approximations

$$y^M(\boldsymbol{x},\boldsymbol{u};t) = \sum_{i\in I} w_i^M(\boldsymbol{x},\boldsymbol{u})\Psi_i(t)$$

$$y_r^F(\boldsymbol{x};t) = \sum_{i\in I} w_{ir}^F(\boldsymbol{x})\Psi_i(t)\,. \tag{3.8}$$

(The accuracy of the approximations using the reduced set of elements for the test bed problem is indicated in Figure 3.2.) We also assume that reality and the

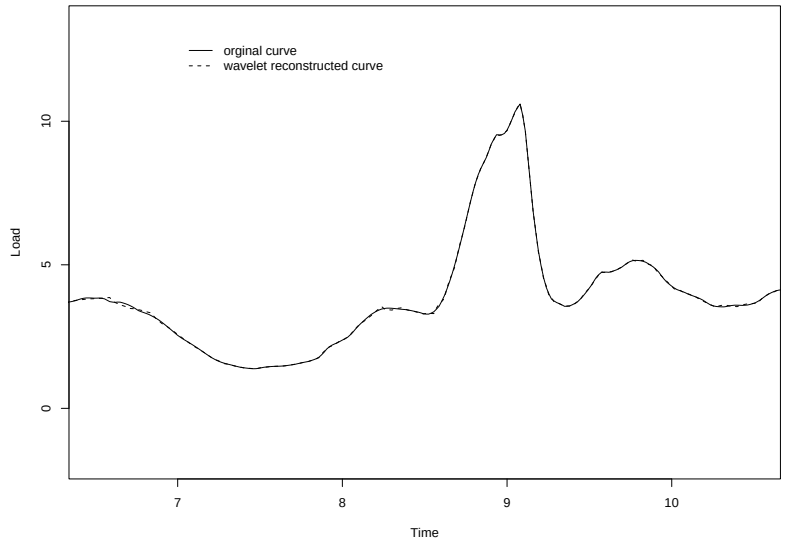

Figure 3.2: The original curve and wavelet reconstructed curve for the first field-run at Site 1.

bias function can be accurately represented by the same basis elements, and write

$$y^R(\boldsymbol{x};t) = \sum_{i\in I} w_i^R(\boldsymbol{x})\Psi_i(t)$$

$$b(\boldsymbol{x};t) = \sum_{i\in I} w_i^b(\boldsymbol{x})\Psi_i(t)\,. \tag{3.9}$$

Matching coefficients and using Equation (3.4) and Equation (3.5), we get

$$w_i^R(\boldsymbol{x}) = w_i^M(\boldsymbol{x},\boldsymbol{u}^*) + w_i^b(\boldsymbol{x}) \quad \forall i \in I\,, \tag{3.10}$$

$$w_{ir}^F(\boldsymbol{x}) = w_i^R(\boldsymbol{x}) + \varepsilon_{ir} \quad \forall i \in I\,. \tag{3.11}$$

We assume that the measurements errors, ε_{ir}, in the wavelet domain are normally distributed with mean zero and are independent across replications r, a standard assumption. However, we also assume that they are independent across i and, indeed, that they are independently normally distributed with mean 0 and possibly differing variances σ_i^2. This independence assumption is needed for the later computations to be feasible, but might seem unrealistic, given that the seven residual functions $(y_r^F(\boldsymbol{x};t) - \bar{y}_r^F(\boldsymbol{x};t))$ – shown in the left hand side of Figure 3.3 – can be seen to be correlated in time t, suggesting that the perhaps the ε_{ir} should be correlated in i.

Interestingly, however, even independent ε_{ir} can lead to correlated error processes, as long as the σ_i^2 differ. Indeed, the error process corresponding to our assumptions on the ε_{ir} is a Gaussian process with mean zero and covariance function $\sum_{i\in I}\sigma_i^2\Psi_i(t)\Psi_i(t')$. The right hand side of Figure 3.3 gives seven realizations of this process, with the σ_i^2 being estimated by the usual unbiased estimates, based on the replicates. The correlation patterns appear to be quite similar between the two processes.

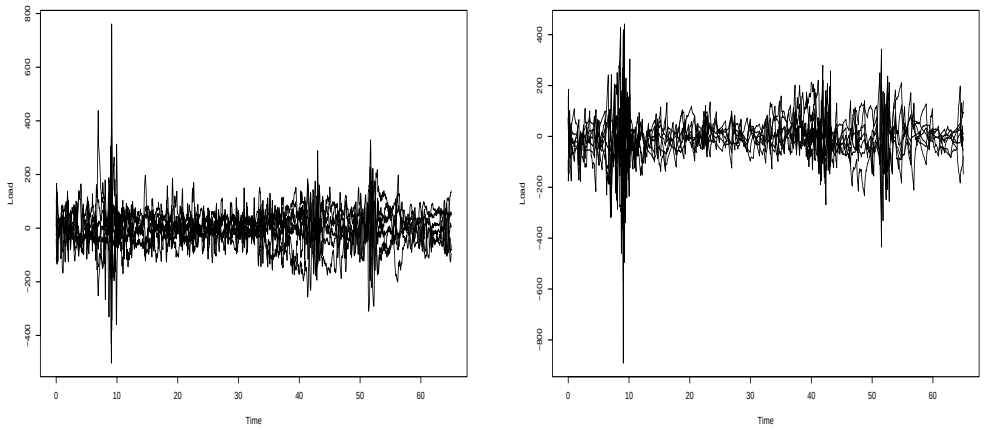

Figure 3.3: The seven residual field processes (left) and seven simulated error processes (right).

Our approach is to analyze each of the retained wavelet coefficients, in equation (3.10) and (3.11), and recombine them to obtain estimates and uncertainties for the "original" functions, in Equation (3.4) and Equation (3.5).

3.2.3 GASP approximation

For y^M, the wavelet coefficients are functions of $(\boldsymbol{x}, \boldsymbol{u})$. Because we cannot freely run the computer model for every $(\boldsymbol{x}, \boldsymbol{u})$ we approximate each of the retained coefficients using data from computer runs. Formally, we start with a Gaussian process prior distribution on a coefficient $w_i^M(\boldsymbol{x}, \boldsymbol{u})$. Given computer model runs, $y^M(\boldsymbol{x}_k, \boldsymbol{u}_k)$, where $\{(\boldsymbol{x}_k, \boldsymbol{u}_k), k = 1, \ldots, K\}$ are the design points in a computer experiment, we extract the data $\{w_i^M(\boldsymbol{x}_k, \boldsymbol{u}_k)\}$ and approximate $w_i^M(\boldsymbol{x}, \boldsymbol{u})$ as the Bayes predictor, the posterior mean, of $w_i^M(\boldsymbol{x}, \boldsymbol{u})$ given the data.

The Gaussian process priors we use are as in the GASP methodology described in Chapter 1: Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$. For each $i \in I$ (the set of retained wavelet

coefficients), the GASP assumption is that $w_i^M(\boldsymbol{z})$ is a Gaussian process with mean μ_i^M, constant variance $1/\lambda_i^M$, and correlation function

$$c_i^M(\boldsymbol{z}, \boldsymbol{z}') = \exp\left(-\sum_{p=1}^{n_M} \beta_{ip}^M |z_p - z_p'|^{2-\alpha_{ip}^M} \right),$$

where n_M is the number of coordinates in $\boldsymbol{z}$, the β's are non-negative parameters and the α's are between 0 and 1.

Let $\boldsymbol{\theta}_i^M = \{\mu_i^M, \lambda_i^M, \alpha_{ip}^M, \beta_{ip}^M; p = 1, \ldots, n_M\}$ be the collection of the (hyper) parameters determining the Gaussian prior distribution of w_i^M. To produce the Bayes predictor we have to deal with the $\boldsymbol{\theta}_i^M$'s; we do so in section 3.3.

3.2.4 Other prior specifications

Priors for $\boldsymbol{u}^*, \boldsymbol{\delta}^*$ are context specific. Engineering advice led to adopting uniform priors for $\boldsymbol{u}^*$ on their ranges in Table 3.1. For the manufacturing variations of the unmeasured parameters in Table 3.1, the advice led to normal priors with standard deviations equal to 1/6 of the ranges of the uncertainty intervals in Table 3.1. Specifically,

$$\pi(u_1) = \pi(u_2) = \text{ Uniform on } [0.125, 0.875]$$

$$\pi(\delta_1) = \pi(\delta_2) \sim N(0, 0.1111^2) \text{ truncated to}[-0.3333, 0.3333]$$

$$\pi(\delta_3) \sim N(0, 0.09723^2) \text{ truncated to } [-0.2917, 0.2917]$$

$$\pi(\delta_4) = \pi(\delta_7) \sim N(0, 0.1026^2) \text{ truncated to}[-0.3077, 0.3077]$$

$$\pi(\delta_5) \sim N(0, 0.04903^2) \text{ truncated to}[-0.1471, 0.1471]$$

$$\pi(\delta_6) \sim N(0, 0.1176^2) \text{ truncated to}[-0.3529, 0.3529]$$

The σ_i^2 are given the usual noninformative priors $\pi(\sigma_i^2) \propto 1/\sigma_i^2$.

Priors for the wavelet coefficients, the $\left(w_i^b\right)$'s, will be Gaussian but restricted to depend only on those coordinates of $\boldsymbol{x}_{nom}$ that vary in the experiments. Because there is only one $\boldsymbol{x}_{nom}$ in the test bed experiment, we only consider the case where the $\left(w_i^b\right)$'s are constants, though a similar approach can be taken for more general settings.

In the wavelet decomposition, each $i \in I$ belongs to some resolution level, $j(i)$(in the test bed – see Appendix B – the levels go from 0 to 12). It is natural and common to model wavelet parameters hierarchically, according to their resolution level. The priors for the biases are then taken as

$$\pi(w_i^b \mid \tau_{j(i)}^2) \sim N\left(0, \tau_{j(i)}^2\right) . \tag{3.12}$$

This is a strong shrinkage prior, shrinking the biases to zero. One might be concerned with such strong shrinkage to zero, but the computer modeling world is one in which biases are typically assumed to be zero, so that utilizing a strong shrinkage prior has the appeal that any detected bias is more likely to be believed to be real in the community than would bias detected with a weaker assumption. (Of course, there are also statistical arguments for using such shrinkage priors.)

The hypervariances τ_j^2 are assigned a variant of a typical objective prior for hypervariances,

$$\pi(\tau_j^2 \mid \{\sigma_i^2\}) \propto \frac{1}{\tau_j^2 + \frac{1}{7}\bar{\sigma}_j^2} ,$$

where $\bar{\sigma}_j^2$ = average of σ_i^2 for i at level j. The $\bar{\sigma}_j^2$ provide a 'scale' for the τ_j^2 and are necessary – or at least some constants in the denominators are necessary – to yield a proper posterior.

3.3 Estimation and Analysis

We restrict attention, for the most part, to the context of the test bed. It will be clear that much of what is done can be generalized.

3.3.1 Approximating wavelet coefficients

In Appendix B we find that, for the test bed, there are 289 wavelet coefficients, w_i^M, to be treated. The Gaussian prior for each has 20 hyperparameters (coordinates of $\boldsymbol{\theta}_i^M$). A full Bayesian treatment would then require treatment of 5780 parameters, an infeasible process. Instead, we treat each coefficient separately and estimate each $\boldsymbol{\theta}_i^M$ by maximum likelihood based on the model-run data $\boldsymbol{w}_i^M = \{w_i^M(\boldsymbol{z}_k)\}$, using code developed by W. Welch. Recall that $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$ and denote the kth design point in the computer experiment by $\boldsymbol{z}_k$. For the test bed there are 65 z_k's.

Letting $\widehat{\boldsymbol{\theta}}_i^M = \{\widehat{\mu}_i^M, \widehat{\lambda}_i^M, \widehat{\alpha}_{ip}^M, \widehat{\beta}_{ip}^M; p = 1, \ldots, n_M\}$ be the maximum likelihood estimates of the $\boldsymbol{\theta}$'s, it follows that the GASP predictive distribution of $w_i^M(\boldsymbol{z})$ at a new $\boldsymbol{z}$ is

$$w_i^M(\boldsymbol{z}) \mid \boldsymbol{w}_i^M, \widehat{\boldsymbol{\theta}}_i^M \sim N(\widehat{m}_i^M(\boldsymbol{z}), \widehat{V}_i^M(\boldsymbol{z}))\,, \tag{3.13}$$

where

$$\widehat{m}_i^M(\boldsymbol{z}) = \widehat{\mu}_i^M + \widehat{\gamma}_i^M(\boldsymbol{z})'(\widehat{\boldsymbol{\Gamma}}_i^M)^{-1}(\boldsymbol{w}_i^M - \widehat{\mu}_i^M \mathbf{1})$$

$$\widehat{V}_i^M(\boldsymbol{z}) = \frac{1}{\widehat{\lambda}_i^M} - \widehat{\gamma}_i^M(\boldsymbol{z})'(\widehat{\boldsymbol{\Gamma}}_i^M)^{-1}\widehat{\gamma}_i^M(\boldsymbol{z})\,,$$

with $\mathbf{1}$ the vector of ones, $\widehat{\boldsymbol{\Gamma}}_i^M$ (65×65 in the test bed example) the covariance matrix for the model-run data $\boldsymbol{w}_i^M$ estimated by plugging-in $\widehat{\boldsymbol{\theta}}_i^M$, and

$$\widehat{\gamma}_i^M(\boldsymbol{z}) = (1/\hat{\lambda}_i^M)(\hat{c}_i^M(\boldsymbol{z}_1, \boldsymbol{z}), \ldots, \hat{c}_i^M(\boldsymbol{z}_k, \boldsymbol{z}))'\,,$$

where $\hat{c}$ is the estimated correlation function. For the rest of the paper we will use equation (3.13) as the definition of the GASP predictive distribution.

Full justification of the use of the plug-in maximum likelihood estimates for the parameters $\boldsymbol{\beta}_i, \boldsymbol{\alpha}_i$ is an open theoretical issue. Intuitively, one expects modest variations in parameters to have little effect on the predictors because they are interpolators. In practice, "studentized" cross-validation residuals (leave-one-out predictions of the data normalized by standard error) have been successfully used to gauge the "legitimacy" of such usage (for examples and additional references see Schonlau and Welch (2005) and Aslett *et al.* (1998)). Only recently, Nagy (2006) has reported simulations indicating reasonably close prediction accuracy of the plug-in MLE predictions to Bayes (Jeffrey priors) predictions in dimensions 1-10 when the number of computer runs = 7×dimension.

All α's and β's are graphed in Figure 3.4. Recall that a β near zero corresponds to a correlation near one, and hence a function that is quite flat in that variable. An α near zero corresponds to a power of two in the exponent of the correlation, suggesting a very smooth functional dependence on that variable. Interestingly, for most of the pairs (α, β), one or the other is near zero.

3.3.2 The posterior distributions

Restricting to the test bed problem we simplify the notation by referring only to $\boldsymbol{\delta}$, the deviation of $\boldsymbol{x}$ from the nominal inputs $\boldsymbol{x}_{nom}$, and rewrite equation (3.10) and (3.11), $\forall i \in I$, as

$$\begin{aligned} w_i^R(\boldsymbol{\delta}^*) &= w_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + w_i^b \\ w_{ir}^F(\boldsymbol{\delta}^*) &= w_i^R(\boldsymbol{\delta}^*) + \varepsilon_{ir}\,, \end{aligned} \tag{3.14}$$

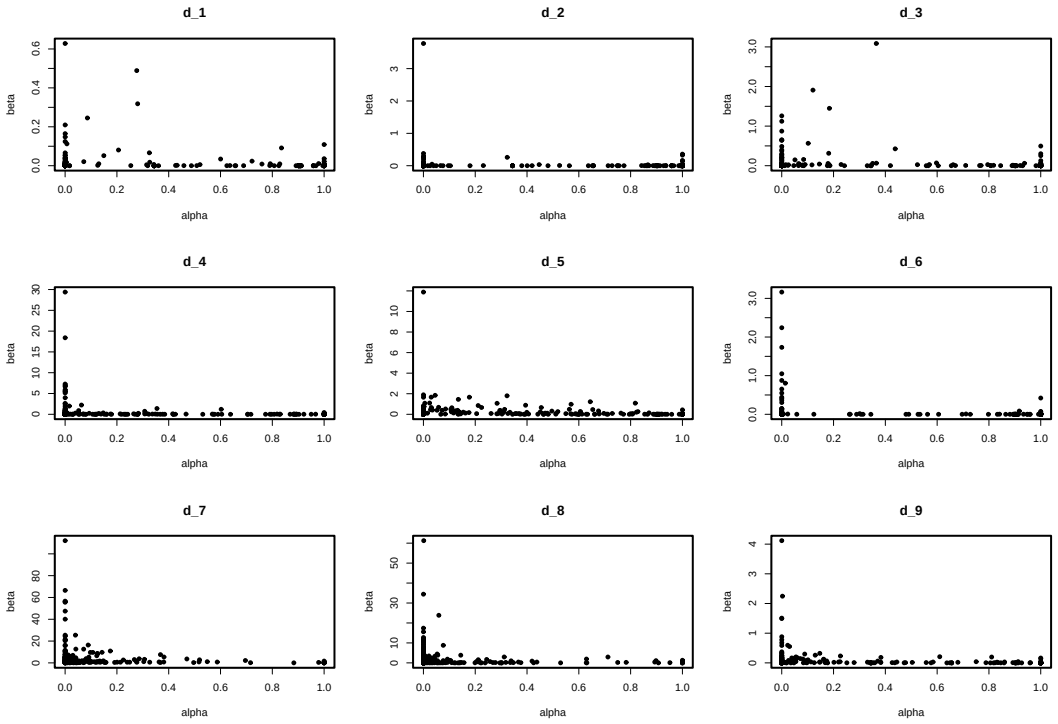

Figure 3.4: (α, β) for all the wavelet coefficients and for each input variable

where the ε_{ir} are independent $\mathrm{N}(0, \sigma_i^2)$.

The field data can be summarized (and simplified) by using the (independent) sufficient statistics

$$\bar{w}_i^F = \frac{1}{7}\sum_{r=1}^{7} w_{ir}^F(\boldsymbol{\delta}^*)\,, \qquad s_i^2 = \sum_{r=1}^{7}(w_{ir}^F(\boldsymbol{\delta}^*) - \bar{w}_i^F)^2\,.$$

(We drop the argument $\boldsymbol{\delta}^*$ for these statistics, since the statistics are actual numbers given from the data.) Key facts to be retained, using equation (3.13) and

(3.14) and properties of normal distributions:

$$\frac{s_i^2}{\sigma_i^2} \mid \sigma_i^2 \sim \chi_6^2\,,$$

$$\bar{w}_i^F \mid w_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*), w_i^b, \sigma_i^2 \sim N\left(w_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + w_i^b, \frac{1}{7}\sigma_i^2\right)\,,$$

$$\bar{w}_i^F \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{w}_i^M, w_i^b, \sigma_i^2 \sim N\left(\widehat{m}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + w_i^b, \widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2\right)\,, \quad (3.15)$$

where (3.13) is used to get the last expression.

Let $\boldsymbol{w}^b$, $\boldsymbol{\tau}^2$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\delta}^*$, and $\boldsymbol{u}^*$ denote the vectors of the indicated parameters and write their prior distribution as

$$\pi(\boldsymbol{w}^b, \boldsymbol{\tau}^2, \boldsymbol{\sigma}^2, \boldsymbol{\delta}^*, \boldsymbol{u}^*) = \pi(\boldsymbol{w}^b \mid \boldsymbol{\tau}^2) \times \pi(\boldsymbol{\tau}^2, \boldsymbol{\delta}^*, \boldsymbol{u}^* \mid \boldsymbol{\sigma}^2) \times \pi(\boldsymbol{\sigma}^2) \quad (3.16)$$

$$= \prod_{i\in I} \pi(\sigma_i^2)\pi\left(w_i^b \mid \tau_{j(i)}^2\right) \times \left[\prod_{j=0}^{12} \pi\left(\tau_j^2 \mid \{\sigma_i^2\}\right) \prod_{i=1}^{7} \pi(\delta_i^*) \prod_{i=1}^{2} \pi(u_i^*)\right]\,.$$

The data, from field and computer model runs, can be summarized as $\boldsymbol{D} = \{\bar{w}_i^F, s_i^2, \boldsymbol{w}_i^M; i = 1, \ldots, 289\}$. Using equation (3.13), (3.15) (3.12) and (3.16), together with standard computations involving normal distributions, it is straightforward to get the posterior distribution of all unknowns as

$$\pi_{post}(w^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*), \boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2 \mid \boldsymbol{D}) = \pi_{post}(w^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) \mid \boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{D})$$

$$\times \pi_{post}(\boldsymbol{w}^b \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \boldsymbol{D}) \times \pi_{post}(\boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\tau}^2 \mid \boldsymbol{\sigma}^2, \boldsymbol{D}) \times \pi_{post}(\boldsymbol{\sigma}^2 \mid \boldsymbol{D})\,, \quad (3.17)$$

where

$$\pi_{post}(w^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) \mid \boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{D}) \sim \prod_{i\in I} N(m_{1i}, V_{1i})\,; \quad (3.18)$$

$$m_{1i} = \frac{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*)}{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\,\sigma_i^2}\,(\bar{w}_i^F - w_i^b) + \frac{\frac{1}{7}\,\sigma_i^2}{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\,\sigma_i^2}\,(\widehat{m}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*))$$

$$V_{1i} = \frac{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*)\,\frac{1}{7}\,\sigma_i^2}{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\,\sigma_i^2}\,,$$

and

$$\pi_{post}(\boldsymbol{w}^b \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \boldsymbol{D}) \sim \prod_{i \in I} N(m_{2i}, V_{2i})\,;$$

$$m_{2i} = \frac{\tau^2_{j(i)}}{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2 + \tau^2_{j(i)}}(\bar{w}_i^F - \widehat{m}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*))\,; \tag{3.19}$$

$$V_{2i} = \frac{\tau^2_{j(i)}(\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2)}{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2 + \tau^2_{j(i)}}\,.$$

The third factor in (3.17) is

$$\pi_{post}(\boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\tau}^2 \mid \boldsymbol{\sigma}^2, \boldsymbol{D}) \propto L(\overline{\boldsymbol{w}}^F, \boldsymbol{s}^2 \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2) \times \pi(\boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\tau}^2 \mid \boldsymbol{\sigma}^2,)\,, \tag{3.20}$$

where the marginal likelihood, L, found by integrating out $\boldsymbol{w}^b$ and $w^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*)$ in the product of the full likelihood and $\pi(\boldsymbol{w}^b \mid \boldsymbol{\tau}^2)$, is

$$L(\overline{\boldsymbol{w}}^F, \boldsymbol{s}^2 \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2) = \prod_{i \in I} \frac{1}{\sqrt{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2 + \tau^2_{j(i)}}} \times \exp\left\{-\frac{1}{2}\left(\frac{(\bar{w}_i^F - \widehat{m}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*))^2}{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2 + \tau^2_{j(i)}}\right)\right\}\,.$$

Finally, the fourth factor in (3.17) is

$$\pi_{post}(\boldsymbol{\sigma}^2 \mid \boldsymbol{D}) \propto \left[\prod_{i \in I} \frac{1}{(\sigma_i^2)^3} \exp\left\{-\frac{s_i^2}{2\sigma_i^2}\right\}\right] \times \int L(\overline{\boldsymbol{w}}^F, \boldsymbol{s}^2 \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2)\; d\boldsymbol{\delta}^*\; d\boldsymbol{u}^*\; d\boldsymbol{\tau}^2\,. \tag{3.21}$$

At this point we make an approximation, and ignore the integral in (3.21); i.e., we simply utilize the replicate observations to determine the posteriors for the σ_i^2. The reason for this is not computational; indeed, one can include the σ_i^2 in

the posterior in (3.20) and deal with them by a Metropolis algorithm. Instead, the motivation is what we call *modularization*, which is meant to indicate that it can be better to separately analyze pieces of the problem than to perform one global Bayesian analysis. The difficulty here is that there is a significant confounding in the posterior distribution between the calibration parameters, the bias function, and the σ_i^2 and this, for instance, allows bias to be replaced by larger σ_i^2. Here we have seven replicate observations for each σ_i^2, so simply utilizing the replicate observation posteriors and preventing the confounding has intuitive appeal. (Making this argument formally is not so easy; in a sense, it occurs because part of the model – here the modeling of the bias – is quite uncertain. Better or more robust modeling of the bias can correct the problem within a full Bayesian analysis, but the difficulty of doing so argues for the simpler modular approach. We will discuss these issues more fully in Section 3.6.)

Simulating from equation (3.18) and (3.19) and the first factor of (3.21) is, of course, trivial, but simulating from equation (3.20) requires MCMC methodology. Given the complexity of the problem, the MCMC requires careful choice of proposal distributions in order to achieve suitable mixing. Discussion of these proposals is relegated to Section 3.3.3 because of the level of detail needed to describe them, but we note that these are technically crucial for the methodology to work and required extensive exploration.

3.3.3 The MCMC algorithm

We first describe the MCMC algorithm being used to draw the posterior samples. The algorithm has the following four steps within each iteration.

Step 1: For $h = 1, \ldots, 1000$, sample the σ_i^{2h} from the following distribution:

$$\text{InverseGamma}\left(3, \frac{2}{s_i^2}\right) \ \left(shape = 3, scale = \frac{2}{s_i^2}\right).$$

Step 2: For $h = 1, \ldots, 1000$, make draws $\boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}, \boldsymbol{\tau}^{2h}$ from the posterior distribution in equation (3.20). (This is complicated – the process is described last.)

Step 3: Given $\boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}, \boldsymbol{\sigma}^{2h}, \boldsymbol{\tau}^{2h}$ draw $\boldsymbol{w}^{bh}$ from the distribution in equation (3.19). (This is simply done by making a draw, for each i, from a normal distribution with the specified means and variances).

Step 4: Given $\boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}, \boldsymbol{\sigma}^{2h}, \boldsymbol{\tau}^{2h}, \boldsymbol{w}^{bh}$, make a draw of w^{Mh} from the distribution in equation (3.18). (Again this is simply done by draws from normal distributions).

For Step 2, we use a Metropolis-Hastings scheme to generate the $(h+1)^{st}$ sample. We break this up into two sub-steps.

Step 2.1 Propose $\boldsymbol{\tau}^2$ by generating from $q(\boldsymbol{\tau}^2 \mid \boldsymbol{\tau}^{2h}) = \prod_{i=0}^{12} q_i(\tau_i^2 \mid \tau_i^{2h})$, where

$$q_i(\tau_i^2 \mid \tau_i^{2h}) \propto \begin{cases} \frac{1}{\tau_i^2} & \text{if } \tau^2 \in [\tau_i^{2h} e^{-0.7}, \tau_i^{2h} e^{0.7}] \\ 0 & \text{otherwise}. \end{cases}$$

The posterior density of $\boldsymbol{\tau}^2$ is not very spiked, so this type of fairly broad local proposal works well. Finally, form the Metropolis-Hastings Ratio

$$\rho = \frac{\pi(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^2 \mid \boldsymbol{\sigma}^{2h}, \boldsymbol{D})\, q(\boldsymbol{\tau}^{2h} \mid \boldsymbol{\tau}^2)}{\pi(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2h} \mid \boldsymbol{\sigma}^{2h}, \boldsymbol{D})\, q(\boldsymbol{\tau}^2 \mid \boldsymbol{\tau}^{2h})}$$

and define $\boldsymbol{\tau}^{2(h+1)} = \boldsymbol{\tau}^2$ with probability $\min(1, \rho)$; $\boldsymbol{\tau}^{2(h+1)} = \boldsymbol{\tau}^{2h}$ otherwise.

Step 2.2 Let $T_k^{\delta} = [a_k^{\delta}, A_k^{\delta}]$ and $T_k^{u} = [a_k^{u}, A_k^{u}]$ denote the intervals on which the prior densities for the corresponding variables are nonzero, and define

$$T_k^{*\delta h} = [\max(a_k^{\delta}, \delta_k^h - 0.05), \min(A_k^{\delta}, \delta_k^h + 0.05)]$$

$$T_k^{*uh} = [\max(a_k^{u}, u_k^h - 0.05), \min(A_k^{u}, u_k^h + 0.05)]\,.$$

Propose $\boldsymbol{\delta}$, $\boldsymbol{u}$ from

$g(\boldsymbol{\delta}, \boldsymbol{u} \mid \boldsymbol{\delta}^h, \boldsymbol{u}^h) =$

$$\prod_{k=1}^{7} \left(\frac{1}{2} U(\delta_k \mid T_k^{\delta}) + \frac{1}{2} U(\delta_k \mid T_k^{*\delta h}) \right) \prod_{k=1}^{2} \left(\frac{1}{2} U(u_k \mid T_k^{u}) + \frac{1}{2} U(u_k \mid T_k^{*uh}) \right) .$$

The logic here is that the posterior densities for some of the parameters is quite flat, so that sampling uniformly over their support (T_k^{δ} or T_k^{u}) would be quite reasonable as a proposal. On the other hand, some of the posteriors are quite concentrated, and for these it is effective to use a locally uniform proposal, centered around the previous value and with a maximum step of 0.05; this leads to uniforms on $T_k^{*\delta h}$ or T_k^{*uh}, which are the regions defined by the intersection of the local uniforms and the support of the priors. Since the goal here was to create a procedure that can be automatically applied for this type of problem, 50-50 mixtures of the two proposals were adopted.

Finally, form the Metropolis-Hastings Ratio

$$\rho = \frac{\pi(\boldsymbol{\delta}, \boldsymbol{u}, \boldsymbol{\tau}^{2(h+1)} \mid \boldsymbol{\sigma}^{2h}, \boldsymbol{D})\, g(\boldsymbol{\delta}^h, \boldsymbol{u}^h \mid \boldsymbol{\delta}, \boldsymbol{u})}{\pi(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2(h+1)} \mid \boldsymbol{\sigma}^{2h}, \boldsymbol{D})\, g(\boldsymbol{\delta}, \boldsymbol{u} \mid \boldsymbol{\delta}^h, \boldsymbol{u}^h)}$$

and set $(\boldsymbol{\delta}^{(h+1)}, \boldsymbol{u}^{(h+1)}) = (\boldsymbol{\delta}, \boldsymbol{u})$ with probability $\min(1, \rho)$, and equal to $(\boldsymbol{\delta}^h, \boldsymbol{u}^h)$ otherwise.

These Metropolis-Hastings steps typically yield highly correlated iterations, so we actually cycle through them 200 times (with fixed $\boldsymbol{\sigma}^{2h}$) before saving the variable values for feeding into Steps 3 and 4.

The end result of the simulation is a sample of draws from the posterior distribution in equation (3.17): each saved draw from the first factor of (3.21) is used to generate the MCMC sample for the third factor, with both being used to generate a draw from the second and the first factors, using equations (3.18) and (3.19). We saved every 200^{th} draw from 200,000 MCMC iterations for the third factor, thereby obtaining a final sample of 1000 draws

$$\{w^{M,h}(\boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}), \boldsymbol{w}^{bh}, \boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}, \boldsymbol{\sigma}^{2h}, \boldsymbol{\tau}^{2h}; h = 1, \ldots, 1000\}\,. \tag{3.22}$$

The results in Section 3.4 are based on this sample from the posterior.

3.4 Results

3.4.1 Estimates of $\boldsymbol{\delta}^*, \boldsymbol{u}^*$

Histograms for $\boldsymbol{\delta}^*, \boldsymbol{u}^*$ (Figure 3.5) are obtained by forming a histogram for each component of $\boldsymbol{\delta}^*, \boldsymbol{u}^*$ from the corresponding elements in equation (3.22). The calibration parameters are moderately affected by the data but, of the input variables, only x_5 and x_6 have posteriors that are significantly different than the priors. The posterior for x_6 is piled up at the end of the allowed range for the variable, which suggests the (undesirable) possibility that this uncertain input is being used as a tuning parameter to better fit the model; a case could be made for preventing this by additional modularization.

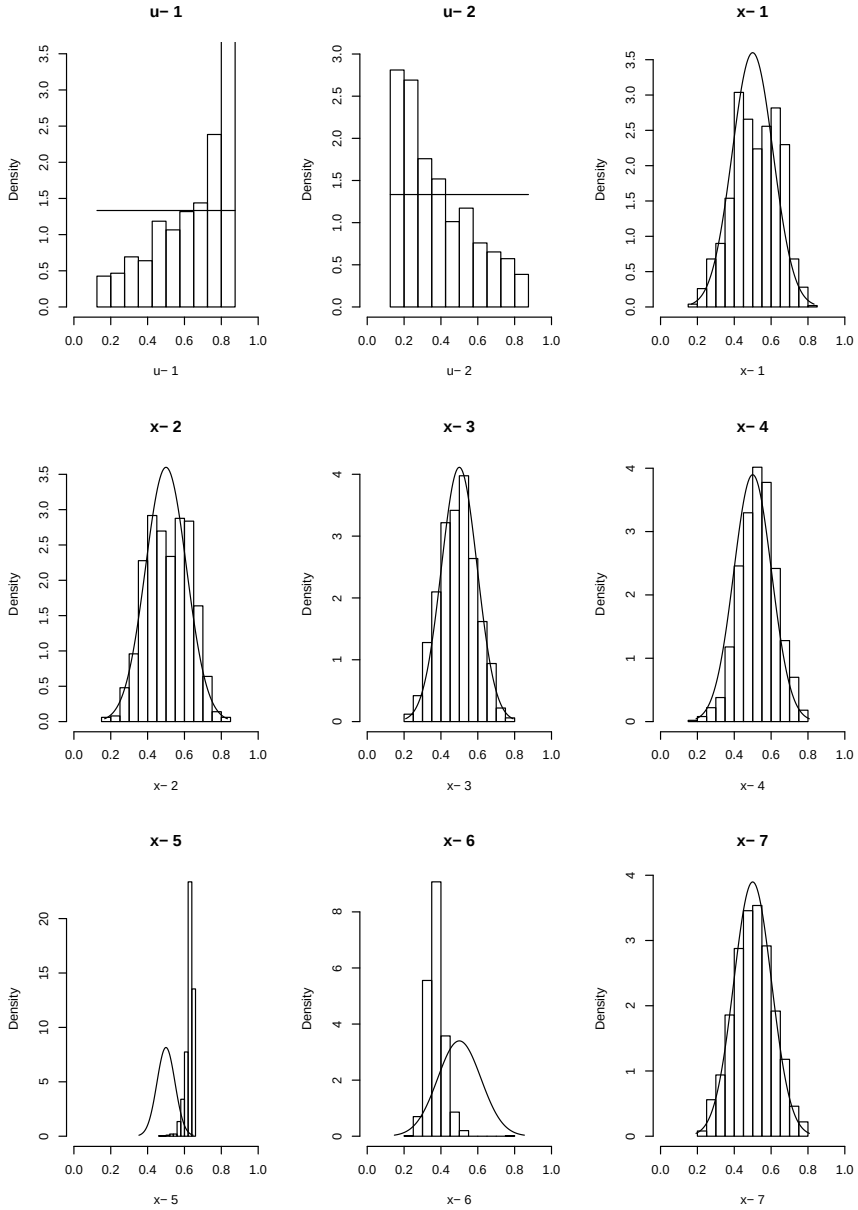

Figure 3.5: Histogram of the posterior draws for the input and calibration parameters with their priors (solid lines).

3.4.2 Estimation of bias and reality

Posterior distributions of the bias and reality curves are obtained by recombining the wavelets with the posterior wavelet coefficients from equation (3.22). For instance, the posterior distribution of b is represented by the sample curves

$$b^h(t) = \sum_{i \in I} w_i^{bh} \Psi_i(t); \quad h = 1, \ldots, 1000\,. \tag{3.23}$$

The posterior mean curve, $\widehat{b}(t) = \frac{1}{1000} \sum_{h=1}^{1000} b^h(t)$, is plotted as the dashed line in Figure 3.6. The uncertainty of this estimate of b is quantified by producing upper and lower uncertainty (tolerance) bounds at each t by, for example, taking the lower $\alpha/2$ and upper $1 - \alpha/2$ quantiles of the posterior distribution of $b(t)$ i.e.,

$$\begin{aligned} L^b(t) &= \frac{\alpha}{2} \text{ quantile of } \{b^h(t); h = 1, \ldots, 1000\} \\ U^b(t) &= \left(1 - \frac{\alpha}{2}\right) \text{ quantile of } \{b^h(t); h = 1, \ldots, 1000\}. \end{aligned} \tag{3.24}$$

These lower and upper bounds are also plotted in Figure 3.6. It is apparent in Figure 3.6 that the bias function is significantly different from 0 especially in the neighborhood of 8.7 and 9.1.

The bounds in equation (3.24) are symmetrically defined. Alternative tolerance bounds can be defined by only requiring that $100\alpha\%$ of the curves lie outside the bounds; a useful choice would satisfy this condition and minimize the width of the bounds: $U^b(t) - L^b(t)$.

Figures 3.5 and 3.6 provide marginal distributions of $\boldsymbol{u}^*, \boldsymbol{x}_{nom} + \boldsymbol{\delta}^*$ and the bias, but it is important to note that these are highly dependent in the posterior. Hence most analyses involving these quantities must be based on their joint, rather than marginal, distributions.

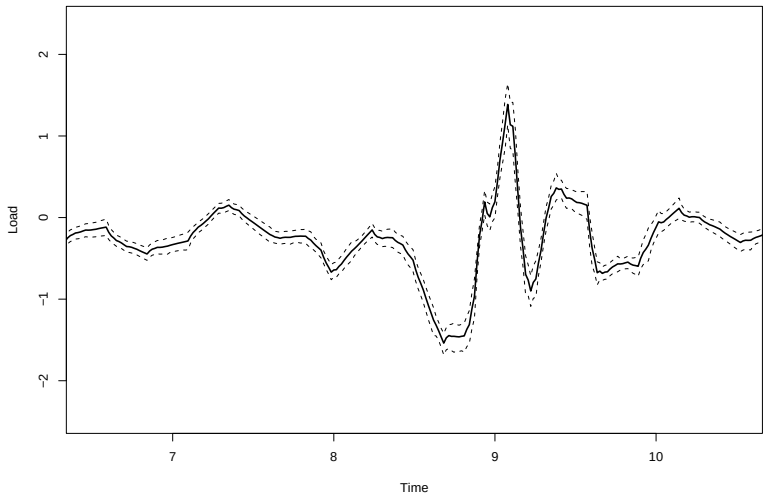

Figure 3.6: The estimate of the bias function (solid line) with 90% tolerance bounds (dashed lines)for Suspension Site 1 and at Region 1.

Estimating reality with uncertainty bounds is done similarly: take the sample of wavelet coefficients $w_i^{Rh} = w_i^{Mh}(\boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}) + w_i^{bh}$ and form

$$y^{Rh}(t) = \sum_i w_i^{Rh} \Psi_i(t), \quad \widehat{y}^R(t) = \frac{1}{1000} \sum_h y^{Rh}(t)\,,$$

$$L^R(t) = \frac{\alpha}{2} \text{ quantile of } \{y^{Rh}(t); h = 1, \ldots, 1000\}\,,$$

$$U^R(t) = \left(1 - \frac{\alpha}{2}\right) \text{ quantile of } \{y^{Rh}(t); h = 1, \ldots, 1000\}\,. \tag{3.25}$$

We call $\widehat{y}^R(t)$ the *bias-corrected prediction of reality.* Figure 3.7 exhibits the bias-corrected prediction and associated uncertainty band.

Figure 3.7 further shows a comparison between bias-corrected prediction and

pure model prediction, the latter being defined as follows:

$$\widehat{y}^M(t) = \sum_i \widehat{m}_i^M \left(\widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}\right) \Psi_i(t)\,, \tag{3.26}$$

where $\widehat{\boldsymbol{\delta}} = \frac{1}{1000}\sum_h \boldsymbol{\delta}^{*h}$ and $\widehat{\boldsymbol{u}} = \frac{1}{1000}\sum_h \boldsymbol{u}^{*h}$ and $\widehat{m}_i^M \left(\widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}\right)$ is the posterior mean of the wavelet coefficients with plugged-in estimates for the unknown parameters (use equation (3.13)). In practice, it may be that running the computer model

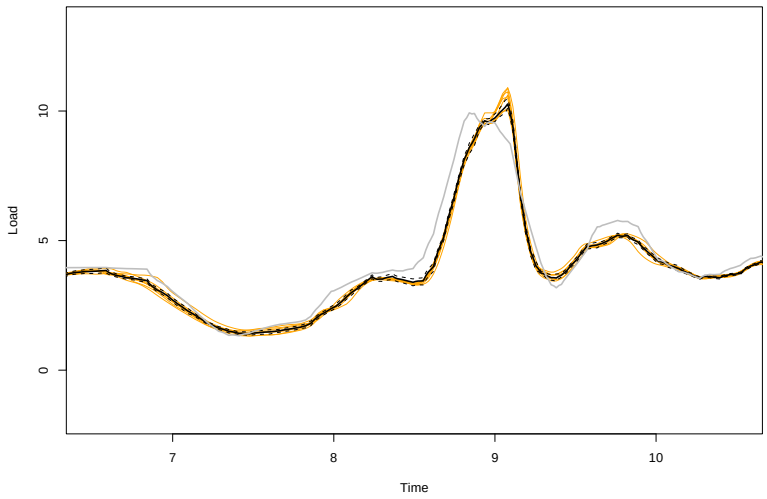

Figure 3.7: Bias-corrected prediction of reality (solid black line) with 90% tolerance bands (dashed black lines), pure model prediction (solid grey line), and field runs (solid yellow lines).

after estimating $\boldsymbol{\delta}^*$, $\boldsymbol{u}^*$ is feasible. Then an alternative (and preferred) pure model prediction is $y^M(\widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}; t)$.

Assessing the uncertainty for predicting reality of the pure model prediction (equation 3.26) (or a new model run producing $y^M(\widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}; t)$) can be done by considering samples $\{y^{Rh}(t) - \widehat{y}^M(t)\}$ or $\{y^{Rh}(t) - y^M(\widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}; t)\}$ and forming bounds.

Here it may be useful to consider asymmetric bounds because the pure model predictions may lie entirely above or below the realizations of reality. But plots like that of Figure 3.7 already show the gap between pure model prediction and reality.

3.4.3 Predicting a new run; same system, same vehicle (new run)

In some prediction settings, not necessarily the one of the test bed, there is interest in predicting a new *field run* with the same inputs (and the same system). Prediction is done by adding in draws, ε_i^h, from a $N(0, \sigma_i^{2h})$ distribution and then following the same prescription as in equation (3.25) to form $\widehat{y}^F(t)$ and corresponding uncertainty bounds. This, of course, produces wider uncertainty bounds.

3.4.4 Extrapolating

There are many follow-on settings where prediction is called for. We single out three such: (i) the same system/vehicle except that $\boldsymbol{x}_{nom}$ changes; (ii) same system/same vehicle type but new components that is, same nominal values and prior distributions for the manufacturing variations but a new realization of the $\boldsymbol{\delta}$'s from the prior distribution; (iii) new system, same vehicle type: different nominal $(x_1, \ldots, x_7)$ inputs, but the same prior distributions for the $\boldsymbol{\delta}$'s.

Any analysis we perform is severely constrained by the fact that we have limited field data on one set of inputs measured with error.

Same system/vehicle, different load

Here the same system was tested with the same vehicle but with added mass. This causes a known change in $\boldsymbol{x}_{nom}$, with everything else (including the $\boldsymbol{\delta}$'s) remaining unchanged. The effect of a simple change in inputs such as this can be addressed using the difference in computer model runs at the *nominal* values of the inputs. More formally, suppose we add a computer model run at $(\boldsymbol{x}_{nom} + \boldsymbol{\Delta}, \boldsymbol{u}_{nom})$ where $\boldsymbol{\Delta}$ is the modest change in the nominal inputs ($\boldsymbol{\Delta}$ has only one non-zero coordinate if the only change is in the mass). Suppose we also have a run at the old nominals $(\boldsymbol{x}_{nom}, \boldsymbol{u}_{nom})$ – if not, use the GASP prediction based on the existing runs. Our assumption then is that (with $\boldsymbol{x}^*$ referring to the true unknown input values for the original system)

$$y^M(\boldsymbol{x}^*+\boldsymbol{\Delta}, \boldsymbol{u}^*; t)-y^M(\boldsymbol{x}^*, \boldsymbol{u}^*; t) \simeq y^M(\boldsymbol{x}_{nom}+\boldsymbol{\Delta}, \boldsymbol{u}_{nom}; t)-y^M(\boldsymbol{x}_{nom}, \boldsymbol{u}_{nom}; t) \equiv \boldsymbol{D}(t).$$

We can then make predictions under the new inputs by simply adding $\boldsymbol{D}(t)$ to the old predictions.

This is illustrated in Figure 3.8; the given bias corrected prediction and tolerance bands for the system with the additional mass is simply the results of Section 3.4.3 translated by $D(t)$. The yellow line is the actual result from a field test of the system with added mass, and the strategy appears successful in the critical Region 1.

New Components in vehicle of same type

In this setting the nominal values $\boldsymbol{x}_{nom}$ remain the same but the new $\boldsymbol{\delta}$'s are random draws from their *prior* (population) distribution and are therefore different than those for the field-tested system. This is of particular interest in practice,

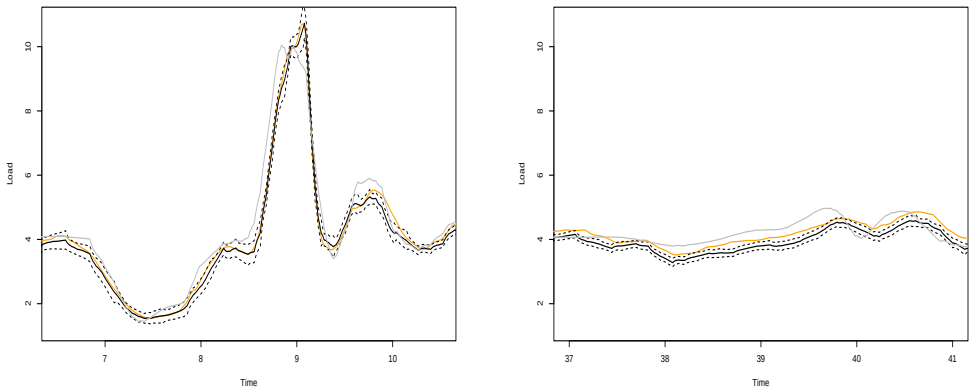

Figure 3.8: Prediction at Site 1 for both Regions 1 and 2 for a run with additional mass.

in that it is prediction for the population of vehicles of the given type that is of prime interest, rather than just prediction for the single system/vehicle tested.

The calibration parameters $\boldsymbol{u}^*$ do not change; they belong to the model and, if physically real, are inherently the same for all systems/vehicles of the same type. Denote the parameters of the new components by $\boldsymbol{z}_{new} = (\boldsymbol{x}_{nom} + \boldsymbol{\delta}_{new}, \boldsymbol{u}^*)$. The input parameters of the computer runs remain $\boldsymbol{z}_k = (\boldsymbol{x}_{nom} + \boldsymbol{\delta}_k, \boldsymbol{u}_k)$ and $\boldsymbol{z}^* = (\boldsymbol{x}_{nom} + \boldsymbol{\delta}^*, \boldsymbol{u}^*)$ are the true values for the tested system. Denote the associated model wavelet coefficients for the new components by $w^M(\boldsymbol{z}_{new})$.

Since $\boldsymbol{\delta}_{new}$ is independent of $(\boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \boldsymbol{D})$, the predictive (posterior)

distribution is (with the L_i's denoting likelihood terms arising from the data)

$$\begin{aligned}
&\pi_{post}(w^M(\boldsymbol{z}_{new}),\, w^M(\boldsymbol{z}^*), \boldsymbol{w}^b, \boldsymbol{\delta}_{new}, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2 \mid \boldsymbol{D}) \\
&\quad \propto \pi(\boldsymbol{\delta}_{new})\pi(\boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2) L_2(\{\bar{w}_i\}, \{s_i^2\} \mid w^M(\boldsymbol{z}^*), \boldsymbol{w}^b, \boldsymbol{\sigma}^2) \\
&\qquad \times L_1(w^M(\boldsymbol{z}_{new}), w^M(\boldsymbol{z}^*), \{w_i^M(z_k)\} \mid \boldsymbol{\delta}_{new}, \boldsymbol{\delta}^*, \boldsymbol{u}^*) \\
&\quad \propto \pi_{post}(w^M(\boldsymbol{z}_{new}) \mid w^M(\boldsymbol{z}^*), \{w_i^M(z_k)\}, \boldsymbol{\delta}_{new}, \boldsymbol{\delta}^*, \boldsymbol{u}^*) \pi(\boldsymbol{\delta}_{new}) \\
&\qquad \times\ \pi(\boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2) L_2(\{\bar{w}_i\}, \{s_i^2\} \mid w^M(\boldsymbol{z}^*), \boldsymbol{w}^b, \boldsymbol{\sigma}^2) \\
&\qquad \times L_3(w^M(\boldsymbol{z}^*), \{w_i^M(z_k)\} \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*)\,. \qquad (3.27)
\end{aligned}$$

To sample from equation (3.27), note that the last three factors in the expression yield exactly the same posterior for $(w^M(\boldsymbol{z}^*), \boldsymbol{w}^b, \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2)$ as before (with the same modularization used), so the draws from the previous MCMC can be used in the new computations. Since $\boldsymbol{\delta}_{new}$ can be drawn from its prior $\pi(\boldsymbol{\delta}_{new})$, it only remains to draw from $\pi_{post}(w^M(\boldsymbol{z}) \mid w^M(\boldsymbol{z}^*), \{w_i^M(z_k)\}, \boldsymbol{\delta}_{new}, \boldsymbol{\delta}^*, \boldsymbol{u}^*)$. But this is simply the GASP distribution where $w^M(\boldsymbol{z}^*)$ has to be added to the model run data. Therefore one simply determines the GASP for the augmented runs $\boldsymbol{w}_i^{M0} = (w_i^M(\boldsymbol{z}_1), w_i^M(\boldsymbol{z}_2), \ldots, w_i^M(\boldsymbol{z}_k), w_i^M(\boldsymbol{z}^*))$, i.e.

$$w_i^M(\boldsymbol{z}_{new}) \mid \boldsymbol{w}_i^{M0}, \widehat{\boldsymbol{\theta}}_i^M \sim N(\widehat{m}_i^{M0}(\boldsymbol{z}_{new}), \widehat{V}_i^{M0}(\boldsymbol{z}_{new}))\,, \qquad (3.28)$$

where $\widehat{\boldsymbol{\theta}}_i^M$ is as in Section 3.3 and

$$\begin{aligned}
\widehat{m}_i^{M0}(\boldsymbol{z}_{new}) &= \widehat{\mu}_i^M + \widehat{\gamma}_i^{M0}(\boldsymbol{z}_{new})'(\widehat{\boldsymbol{\Gamma}}_i^{M0})^{-1}(\boldsymbol{w}_i^{M0} - \widehat{\mu}_i^M \mathbf{1}) \\
\widehat{V}_i^{M0}(\boldsymbol{z}_{new}) &= \frac{1}{\widehat{\lambda}_i^M} - \widehat{\gamma}_i^{M0}(\boldsymbol{z}_{new})'(\widehat{\boldsymbol{\Gamma}}_i^{M0})^{-1}\widehat{\gamma}_i^{M0}(\boldsymbol{z}_{new})\,,
\end{aligned}$$

where $\widehat{\gamma}_i^{M0}(\boldsymbol{z}_{new}) = (1/\hat{\lambda}_i^M)(\hat{c}_i^M(\boldsymbol{z}_1, \boldsymbol{z}_{new}), \ldots, \hat{c}_i^M(\boldsymbol{z}_k, \boldsymbol{z}_{new}), \hat{c}_i^M(\boldsymbol{z}^*, \boldsymbol{z}_{new}))'$ and $\widehat{\boldsymbol{\Gamma}}_i^{M0}$ is obtained by appending the column $\widehat{\gamma}_i^{M0}(\boldsymbol{z}^*)$ and row $\widehat{\gamma}_i^{M0}(\boldsymbol{z}^*)'$ to $\widehat{\boldsymbol{\Gamma}}_i^M$. Note that

to calculate $(\widehat{\mathbf{\Gamma}}_i^{M0})^{-1}$, one can utilize the standard updating formula

$$\begin{bmatrix} 1 & a \\ a^T & \Gamma \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{q} & -\frac{1}{q}a^T\Gamma^{-1} \\ -\frac{1}{q}\Gamma^{-1}a & \Gamma^{-1} + \frac{1}{q}\Gamma^{-1}aa^T\Gamma^{-1} \end{bmatrix}$$

where $q = 1 - a^T\Gamma^{-1}a$.

Application of these expressions yield draws $h = 1, \ldots, 1000$ from the posterior distribution of the i^{th} wavelet coefficient for the new system as

$$w_i^{Fh}(\boldsymbol{z}^h) = w_i^M(\boldsymbol{z}^h) + w_i^{bh} + \varepsilon_i^h$$

where $\varepsilon_i^h \sim N(0, \sigma_i^{2h})$. Figure 3.9 plots the predictions of the new system with uncertainty bands. The uncertainty has increased because the prior for $\boldsymbol{\delta}_{new}$ is used rather than the posterior for the tested system, compare with Figure 3.7.

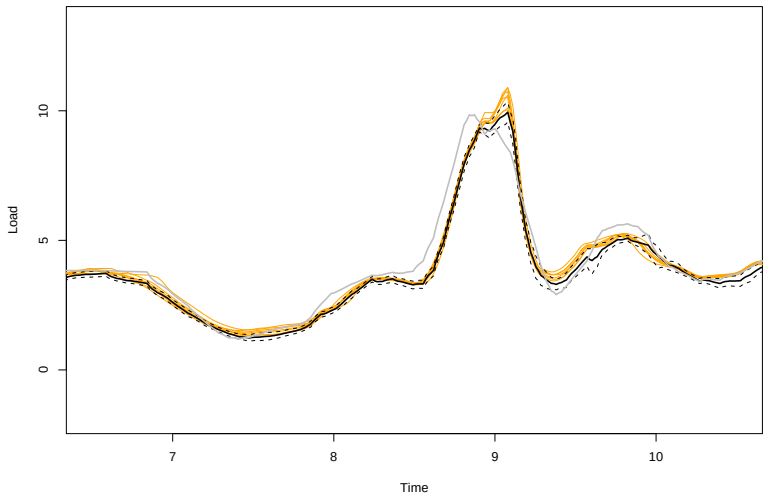

Figure 3.9: Predictions for a new components, same vehicle type.

New vehicle with new nominals

The primary engineering use of computer models is to extrapolate to a system with new nominals when there is no new field data. This will require strong assumptions, especially about the bias. The simplest assumption about the bias, which we make here, is that the new system has the same bias function as the old system. The calibration parameters $\boldsymbol{u}^*$ are also assumed to remain the same. We use the joint – and highly dependent – posterior distribution of the bias and $\boldsymbol{u}^*$ from the original system extensively in what follows.

The new system has the same I/U map as the original system, but with new nominal values $\boldsymbol{x}^B_{nom}$ (which we refer to as Condition B). The new $\boldsymbol{\delta}$'s are taken to have the same priors as for Condition A. The same 65 point design on $(\boldsymbol{\delta}, \boldsymbol{u})$ was used as before with the addition of one central value. Again one run failed, leaving 65 usable model runs. Registration of the output curves is unnecessary because there are no field data and the computer runs are assumed to be inherently registered. The resulting computer runs were passed through the same wavelet decomposition as before retaining only those coefficients that appeared earlier. The resulting GASP for the i^{th} wavelet coefficient is

$$w_i^{BM}(\boldsymbol{z}) \mid \boldsymbol{w}_i^{BM}, \widehat{\boldsymbol{\theta}}_i^{BM} \sim N\left(\widehat{m}_i^{BM}(\boldsymbol{z}), \widehat{V}_i^{BM}(\boldsymbol{z})\right) \tag{3.29}$$

exactly as in equation (3.13).

This new GASP analysis is done only with Condition B data. A GASP analysis combining the model runs for Condition B *and* the model runs for the original system is not used because the changes in the nominals are too large to safely assume connections between the computer model runs for the two systems.

The situation now is analogous to that of the previous argument for new

vehicles of the same type with the same nominals. In the current case, again using the independence of $\boldsymbol{\delta}^B$ from the other unknowns, the predictive (posterior) distribution of the relevant unknowns can be written as

$$\begin{aligned}
&\pi(\boldsymbol{w}^{BM}(\boldsymbol{z}^B), \boldsymbol{w}^b, \boldsymbol{\delta}^B, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2 \mid \boldsymbol{D}) \\
&\qquad = \pi(\boldsymbol{w}^{BM}(\boldsymbol{z}^B) \mid \boldsymbol{w}^b, \boldsymbol{\delta}^B, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \boldsymbol{D}) \times \pi(\boldsymbol{\delta}^B) \times \pi(\boldsymbol{w}^b, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2 \mid \boldsymbol{D}) \\
&\qquad = \pi(\boldsymbol{w}^{BM}(\boldsymbol{z}^B) \mid \boldsymbol{\delta}^B, \boldsymbol{u}^*, \boldsymbol{D}) \times \pi(\boldsymbol{\delta}^B) \times \pi(\boldsymbol{w}^b, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2 \mid \boldsymbol{D})\,;
\end{aligned}$$

here $\pi(\boldsymbol{w}^{BM}(\boldsymbol{z}^B) \mid \boldsymbol{\delta}^B, \boldsymbol{u}^*, \boldsymbol{D})$ is just the GASP distribution in equation (3.29) and $\pi(\boldsymbol{\delta}^B)$ is the prior for B inputs from the I/U map. Draws of $\boldsymbol{w}^b, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2$ are made from the old posterior distribution for the original system. Because $\boldsymbol{w}^b$ and $\boldsymbol{u}^*$ are highly dependent in the posterior, they must be jointly sampled for the extrapolation; naive approaches – such as simply trying to add the bias from Figure 3.6 to the pure model prediction – will not succeed.

The "carry-over" assumptions for the bias and the field variances lead to draws from the posterior distribution of the wavelet coefficients for B to be

$$w_i^{BFh}(\boldsymbol{z}^{Bh}) = w_i^{BM}(\boldsymbol{z}^h) + w_i^{bh} + \varepsilon_i^h$$

where $\varepsilon_i^h \sim N(0, \sigma_i^{2h})$.

In the left of Figure 3.10 the prediction for B is presented. Actual field data (8 replicate runs for B) were afterwards available (not used in constructing the predictions and tolerance bands) and they are superimposed on the plots in the left of Figure 3.10. The effectiveness of carrying over the assumptions from A to B is apparent. If such strong assumptions, as the constancy of bias, are to be made it is best to be extremely careful about implementing the assumption. Here, for instance, physics considerations might suggest that an assumption of constant

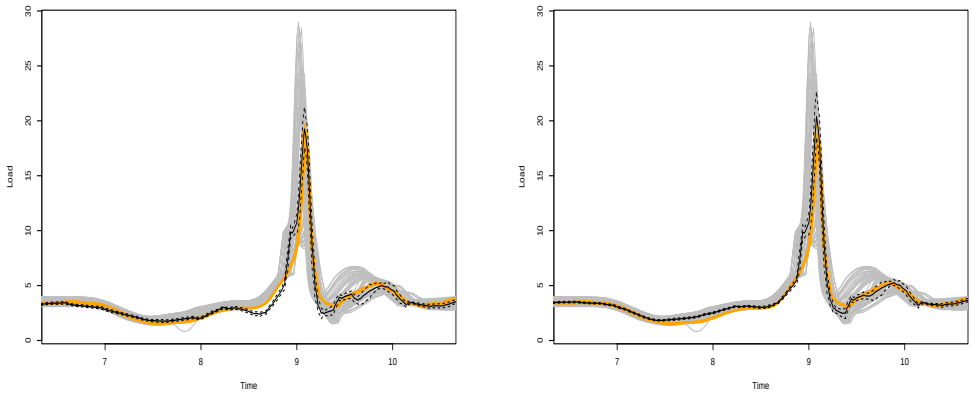

Figure 3.10: Prediction at Site 1 of a new system under Condition B in Region 1. Left: additive bias. Right: multiplicative bias.

multiplicative bias might be more sensible than an assumption of constant additive bias. The standard way of implementing this would be to analyze the log of the output data. A simpler alternative is to transform the additive biases obtained above into multiplicative biases, and apply these multiplicative biases to the GASP draws under Condition B. Bias in the additive system can be written

$$b^h(t) = y^{Rh}(t) - y^{Mh}(t)\,;$$

the corresponding multiplicative representation of the bias is

$$b^h_{mult}(t) = \frac{y^{Rh}(t)}{y^{Mh}(t)} - 1\,,$$

which would lead to draws from the posterior for reality under Condition B of

$$y^{BRh}(t) = y^{BMh}(t) \times (1 + b^h_{mult}(t))\,.$$

The right of Figure 3.10 presents the analogue of the left using the multiplicative bias. The additive and multiplicative predictions are not notably different;

the edge going to the additive predictions at the high peak and otherwise to the multiplicative predictions. The next section discusses another site in the suspension system for which the analysis in the paper was implemented, called Site 2. For this site, the multiplicative predictions under Condition B were significantly better than the additive predictions, as indicated in Figure 3.11.

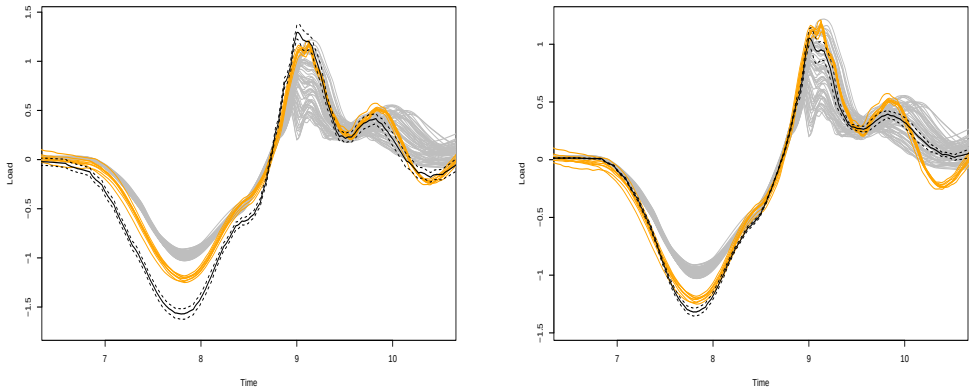

Figure 3.11: Prediction at Site 2 of a new system under Condition B in Region 1. Left: additive bias. Right: multiplicative bias.

3.5 Site 2 Analyses

Analyses for Site 2 of the system proceed in exactly the same way as those for Site 1. The posterior distributions for the calibration parameters (u_1, u_2) as well as for $\boldsymbol{\delta}$ are in Figure 3.12. These are somewhat different than those for Site 1 in Figure 3.5. These parameters are, of course, the same for either site, but the limited data available at each site lead to somewhat different posterior distributions. Also, separately analyzing the sites can result in over-tuning while accommodating the biases at each site individually. A natural solution is to do a bivariate functional

analysis of the two sites jointly. This is being pursued separately.

Figure 3.14 presents the bias corrected predictions, together with tolerance bounds, along with the pure model prediction; other figures for Site 2 are omitted since, with the exception of Figure 3.11, they do not provide further insight.

3.6 Modularization

As we have discussed in Section 3.3.2, there is significant confounding in the posterior distribution between the calibration parameters, the bias function, and the σ_i^2, which allows bias to be replaced by larger σ_i^2. To illustrate this, we perform a global Bayesian analysis, dealing with the σ_i^2's by a Metropolis algorithm. As we see from Figure 3.15, the bias function given by the global Bayesian analysis is shrunk much more toward 0 than that in the modular analysis in Figure 3.6. Consequently, as shown in Figure 3.16, the posterior mean for the prediction of reality is much closer to the computer model runs and consequently has much wider confidence bands than those for the modular analysis in Figure 3.7.

To understand the situation, it is useful to consider a simplified situation - the random effects model

$$y_{ir} = \mu_i + b_i + \epsilon_{ir}, i = 1, \ldots, K; r = 1, \ldots, n\,,$$

where $\boldsymbol{\mu} = \{\mu_i\}$ are assumed known, $b_i \sim \mathrm{N}\,(0, \tau^2)$, and $\epsilon_{ir} \sim \mathrm{N}\,(0, \sigma_i^2)$. We can write the likelihood as

$$\mathrm{L}(\tau^2, \left\{\sigma_i^2\right\}; \left\{\bar{y}_i, s_i^2\right\}) \propto \prod_{i=1}^{K} \frac{\sigma_i^{1-n}}{\sqrt{\tau^2 + \frac{1}{n}\sigma_i^2}} \exp\left(-\frac{\left(\bar{y}_i - \mu_i\right)^2}{2\left(\tau^2 + \frac{1}{n}\sigma_i^2\right)} - \frac{s_i^2}{2\sigma_i^2}\right) \qquad (3.30)$$

where $\bar{y}_i$ and s_i^2 are sufficient statistics, $\bar{y}_i = \sum_{r=1}^{n} y_{ir}$, $s_i^2 = \sum_{r=1}^{n} \left(y_{ir} - \bar{y}_i\right)^2$.

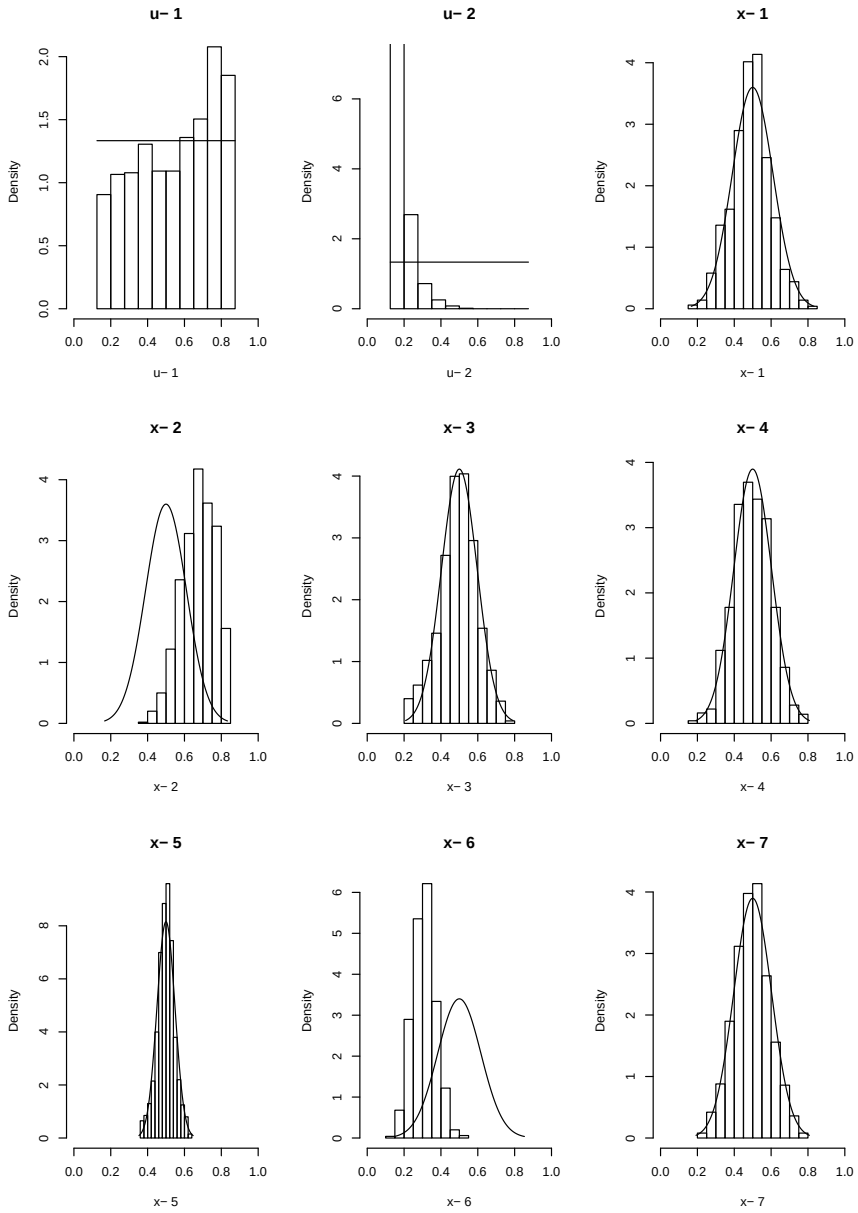

Figure 3.12: Posterior distributions for the unknown input parameters for Site 2 of the system.

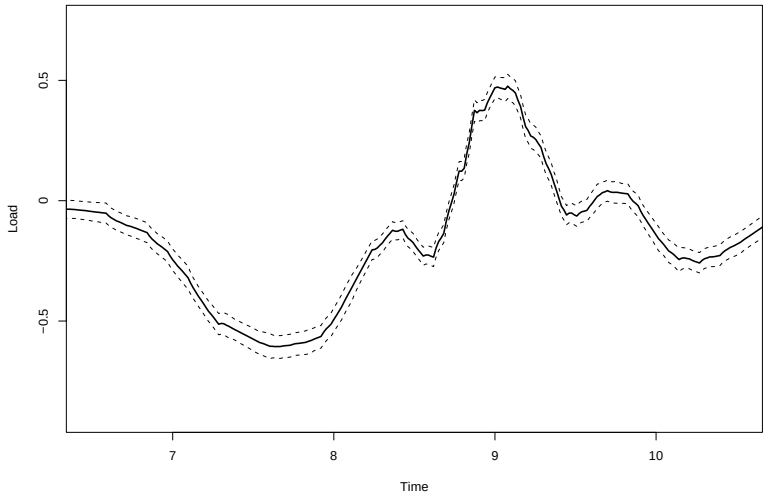

Figure 3.13: Bias estimate with 90% tolerance bounds for Site 2 at Region 1.

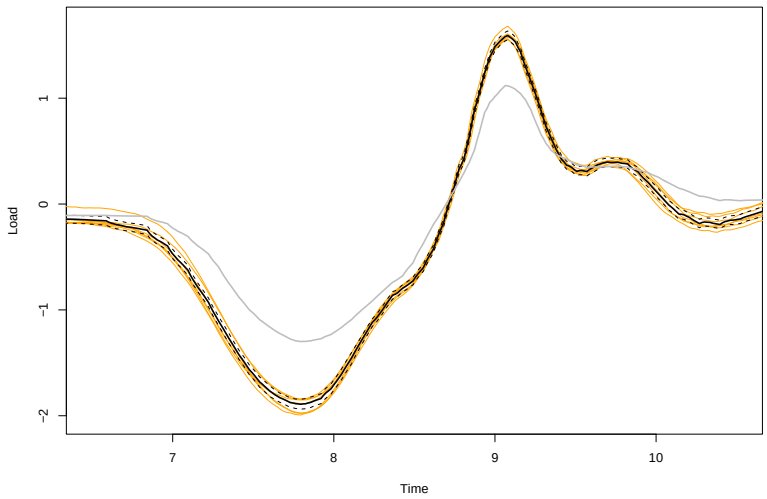

Figure 3.14: Bias-corrected prediction of reality with 90% tolerance bands for Site 2 at Region 1.

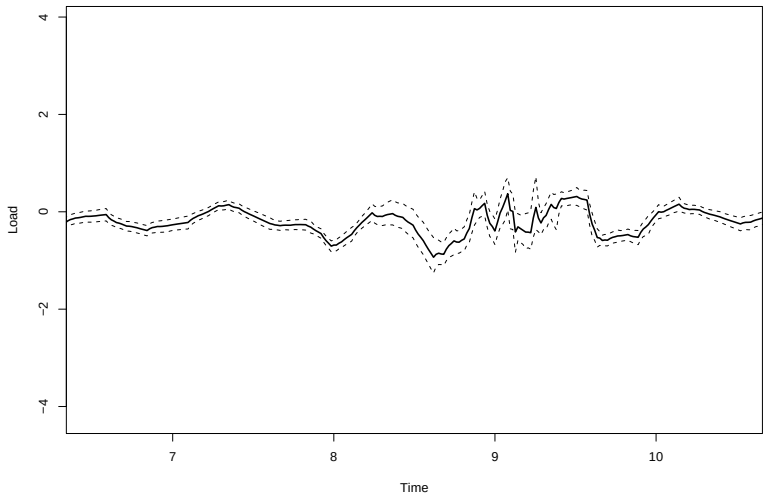

Figure 3.15: Posterior mean of the bias function with 90% confidence bands under the full Bayes approach Normal model assumption.

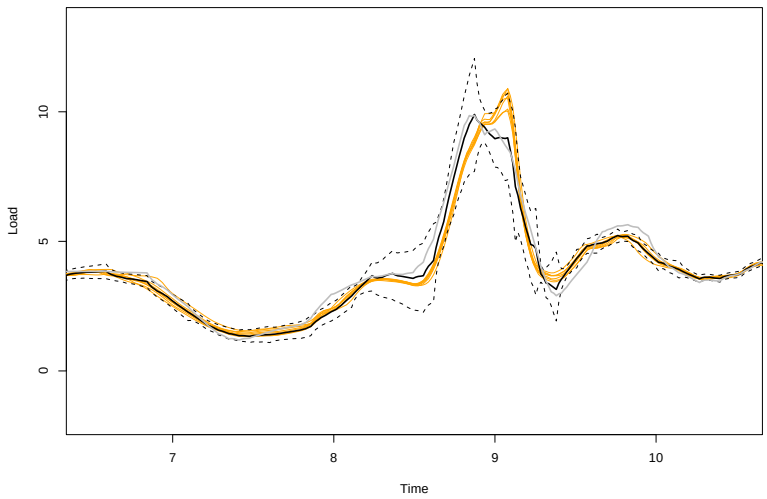

Figure 3.16: Posterior mean of the prediction for reality with 90% confidence bands under the full Bayes approach with Normal model assumption.

Suppose now that $(\bar{y}_i - \mu_i)^2$ is large for some i, as would be true if b_i were an "outlier" (as could be expected for the biases arising in the testbed problem). One usually expects this to be at least partially accommodated by the analysis gravitating to large values of τ^2, the variance of the b_i. However, if K is large, Equation (3.30) suggests that large values of σ_i^2 will result instead. Indeed, in the factor before the exponent in Equation (3.30), the order of τ is τ^{-K}, while that of σ_i^2 is σ_i^{-n}. When K is large and n small, as in the testbed example, it follows that the outliers effect in the exponent will be accommodated by making σ_i^2 large, rather than τ^2. Equation (3.19) shows that the corresponding bias b_i will then be greatly shrunk towards 0. Therefore, we use *modularization* to separate the pieces of the problem, i.e., making inference about the $\{\sigma_i^2\}$ only from the replicate observations.

Another solution to this confounding problem is to use robust distributions for the biases. Thus we consider modeling the biases in Equation (3.12) by the Cauchy distribution

$$\pi(w_i^b \mid \tau_{j(i)}^2) \sim \text{Cauchy}\left(0, \tau_{j(i)}^2\right) . \tag{3.31}$$

We can write the Cauchy distribution as a normal mixture, i.e.,

$$\pi(w_i^b \mid \tau_{j(i)}^2, \lambda_i) \sim \text{Cauchy}\left(0, \tau_{j(i)}^2/\lambda_i\right) , \tag{3.32}$$

with $\lambda_i \sim \text{Gamma}(\frac{1}{2}, 2)$. Letting $\boldsymbol{\lambda} = \{\lambda_i\}$, we have

$$\begin{aligned} \pi(\boldsymbol{w}^b, \boldsymbol{\lambda}, \boldsymbol{\tau}^2, \boldsymbol{\sigma}^2, \boldsymbol{\delta}^*, \boldsymbol{u}^*) =& \pi(\boldsymbol{w}^b \mid \boldsymbol{\tau}^2, \boldsymbol{\lambda})\pi(\boldsymbol{\tau}^2, \boldsymbol{\delta}^*, \boldsymbol{u}^* \mid \boldsymbol{\sigma}^2)\pi(\boldsymbol{\sigma}^2)\pi(\boldsymbol{\lambda}) \\ =& \prod_{i \in I} \pi(\sigma_i^2)\pi(\lambda_i)\pi\left(w_i^b \mid \lambda_i, \tau_{j(i)}^2\right) \\ & \times \left[\prod_{j=0}^{12} \pi\left(\tau_j^2 \mid \{\sigma_i^2\}\right) \prod_{i=1}^{7} \pi(\delta_i^*) \prod_{i=1}^{2} \pi(u_i^*)\right] . \end{aligned} \tag{3.33}$$

Therefore the posterior distribution of all unknowns is

$$\begin{aligned}\pi_{post}(w^M(\boldsymbol{\delta}^*,\boldsymbol{u}^*),&\boldsymbol{w}^b,\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\lambda},\boldsymbol{\tau}^2 \mid \boldsymbol{D})\\ &=\pi_{post}(w^M(\boldsymbol{\delta}^*,\boldsymbol{u}^*) \mid \boldsymbol{w}^b,\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\lambda},\boldsymbol{\tau}^2\boldsymbol{D})\\ &\quad\times \pi_{post}(\boldsymbol{w}^b,\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\lambda},\boldsymbol{\tau}^2 \mid \boldsymbol{D})\,, \end{aligned} \tag{3.34}$$

where $\pi_{post}(w^M(\boldsymbol{\delta}^*,\boldsymbol{u}^*) \mid \boldsymbol{w}^b,\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{D})$ is given in Equation (3.18). Sampling from the second factor in Equation (3.34) is done by a Gibbs sampler. Grouping the parameters as $\{\boldsymbol{w}^b,\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\tau}^2\}$, $\{\boldsymbol{\lambda}\}$, We have

$$\pi_{post}\left(\lambda_i \mid \boldsymbol{w}^b,\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\tau}^2,\boldsymbol{D}\right) \propto \exp\left(-\frac{\tau^2_{j(i)}+(w^b_i)^2}{2\tau^2_{j(i)}}\lambda_i\right)\,, \tag{3.35}$$

and

$$\begin{aligned}\pi_{post}(\boldsymbol{w}^b,\boldsymbol{\delta}^*,&\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\tau}^2 \mid \boldsymbol{\lambda},D) \\ &=\pi_{post}(\boldsymbol{w}^b \mid \boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\lambda},\boldsymbol{\tau}^2,\boldsymbol{D})\pi_{post}(\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\tau}^2 \mid \boldsymbol{\lambda},\boldsymbol{D})\,. \end{aligned} \tag{3.36}$$

The first factor in Equation (3.36) is

$$\pi_{post}(\boldsymbol{w}^b \mid \boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\lambda},\boldsymbol{\tau}^2,\boldsymbol{D}) \sim \prod_{i\in I} N(m_{2i},V_{2i})\,; \tag{3.37}$$

$$m_{2i} = \frac{\tau^2_{j(i)}/\lambda_i}{\widehat{V}^M_i(\boldsymbol{\delta}^*,\boldsymbol{u}^*)+\frac{1}{7}\sigma^2_i+\tau^2_{j(i)}/\lambda_i}(\bar{w}^F_i-\widehat{m}^M_i(\boldsymbol{\delta}^*,\boldsymbol{u}^*))\,,$$

$$V_{2i} = \frac{\tau^2_{j(i)}/\lambda_i(\widehat{V}^M_i(\boldsymbol{\delta}^*,\boldsymbol{u}^*)+\frac{1}{7}\sigma^2_i)}{\widehat{V}^M_i(\boldsymbol{\delta}^*,\boldsymbol{u}^*)+\frac{1}{7}\sigma^2_i+\tau^2_{j(i)}/\lambda_i}\,.$$

The second factor in Equation (3.36) is

$$\begin{aligned}\pi_{post}(\boldsymbol{\delta}^*,\boldsymbol{u}^*,&\boldsymbol{\tau}^2,\boldsymbol{\sigma}^2 \mid \boldsymbol{\lambda},\boldsymbol{D}) \\ &\propto L^c(\overline{\boldsymbol{w}}^F,\boldsymbol{s}^2 \mid \boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\sigma}^2,\boldsymbol{\tau}^2,\boldsymbol{\lambda})\times\pi(\boldsymbol{\delta}^*,\boldsymbol{u}^*,\boldsymbol{\tau}^2,\boldsymbol{\sigma}^2 \mid \boldsymbol{\lambda})\,, \end{aligned} \tag{3.38}$$

where the marginal likelihood under the Cauchy assumption and given $\boldsymbol{\lambda}$, L^c, is

$$L^c(\overline{\boldsymbol{w}}^F, \boldsymbol{s}^2 \mid \boldsymbol{\delta}^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2, \boldsymbol{\lambda}) = \prod_{i \in I} \frac{1}{\sqrt{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2 + \tau_{j(i)}^2/\lambda_i}} \times \exp\left\{ -\frac{1}{2} \left(\frac{(\bar{w}_i^F - \widehat{m}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*))^2}{\widehat{V}_i^M(\boldsymbol{\delta}^*, \boldsymbol{u}^*) + \frac{1}{7}\sigma_i^2 + \tau_{j(i)}^2/\lambda_i} \right) \right\}.$$

The MCMC algorithm to draw samples under the Cauchy assumption proceeds as follows.

Step 1: For $h = 1, \ldots, 1000$, propose σ_i^2 from the following distribution:

$$\text{InverseGamma}\left(3, \frac{2}{s_i^2}\right) \left(shape = 3, scale = \frac{2}{s_i^2} \right).$$

Calculate the acceptance ratio by

$$\rho = \frac{\pi_{post}(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2h}, \boldsymbol{\sigma}^2 \mid \boldsymbol{\lambda}^h, \boldsymbol{D})}{\pi_{post}(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2h}, \boldsymbol{\sigma}^{2h} \mid \boldsymbol{\lambda}^h, \boldsymbol{D})}$$

and define $\boldsymbol{\sigma}^{2(h+1)} = \boldsymbol{\sigma}^2$ with probability $\min(1, \rho)$; $\boldsymbol{\sigma}^{2(h+1)} = \boldsymbol{\sigma}^{2h}$ otherwise.

Step 2: For $h = 1, \ldots, 1000$, propose $\boldsymbol{\tau}^2$ by generating from $q(\boldsymbol{\tau}^2 \mid \boldsymbol{\tau}^{2h})$ defined as in Equation (3.22) and form the Metropolis-Hastings Ratio

$$\rho = \frac{\pi_{post}(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^2, \boldsymbol{\sigma}^{2(h+1)} \mid \boldsymbol{\lambda}^h, \boldsymbol{D})\, q(\boldsymbol{\tau}^{2h} \mid \boldsymbol{\tau}^2)}{\pi_{post}(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2h}, \boldsymbol{\sigma}^{2(h+1)} \mid \boldsymbol{\lambda}^h, \boldsymbol{D})\, q(\boldsymbol{\tau}^2 \mid \boldsymbol{\tau}^{2h})}$$

and define $\boldsymbol{\tau}^{2(h+1)} = \boldsymbol{\tau}^2$ with probability $\min(1, \rho)$; $\boldsymbol{\tau}^{2(h+1)} = \boldsymbol{\tau}^{2h}$ otherwise.

Step 3: For $h = 1, \ldots, 1000$, propose $\boldsymbol{\delta}, \boldsymbol{u}$ as before, and form the Metropolis-Hastings Ratio

$$\rho = \frac{\pi_{post}(\boldsymbol{\delta}, \boldsymbol{u}, \boldsymbol{\tau}^{2(h+1)}, \boldsymbol{\sigma}^{2(h+1)} \mid \boldsymbol{\lambda}^h, \boldsymbol{D})\, g(\boldsymbol{\delta}^h, \boldsymbol{u}^h \mid \boldsymbol{\delta}, \boldsymbol{u})}{\pi_{post}(\boldsymbol{\delta}^h, \boldsymbol{u}^h, \boldsymbol{\tau}^{2(h+1)}, \boldsymbol{\sigma}^{2(h+1)} \mid \boldsymbol{\lambda}^h, \boldsymbol{D})\, g(\boldsymbol{\delta}, \boldsymbol{u} \mid \boldsymbol{\delta}^{h+1}, \boldsymbol{u}^{h+1})}$$

and define $\boldsymbol{\delta}^{h+1} = \boldsymbol{\delta}$, $\boldsymbol{u}^{h+1} = \boldsymbol{u}$ with probability $\min(1, \rho)$; $\boldsymbol{\delta}^{h+1} = \boldsymbol{\delta}^{h}$, $\boldsymbol{u}^{h+1} = \boldsymbol{u}$ otherwise.

Step 4: Given $\boldsymbol{w}^{bh}, \boldsymbol{\tau}^{2(h+1)}$, draw $\boldsymbol{\lambda}^{h+1}$ according to Equation (3.35).

Step 5: Given $\boldsymbol{\delta}^{*(h+1)}, \boldsymbol{u}^{*(h+1)}, \boldsymbol{\sigma}^{2(h+1)}, \boldsymbol{\lambda}^{h+1}, \boldsymbol{\tau}^{2(h+1)}$ draw $\boldsymbol{w}^{b(h+1)}$ from the distribution in Equation (3.37).

Step 6: Given $\boldsymbol{\delta}^{*(h+1)}, \boldsymbol{u}^{*(h+1)}, \boldsymbol{\sigma}^{2(h+1)}, \boldsymbol{\lambda}^{h+1}, \boldsymbol{\tau}^{2(h+1)}, \boldsymbol{w}^{b(h+1)}$, make a draw of $w^{M(h+1)}$ from the distribution in Equation (3.18).

Again, we cycle through 200 times between Step 1, 2, 3 before saving the final values for feeding into Step 4, 5, and 6. The results in Figure 3.17 and Figure 3.18 are based on this sample from the posterior. The bias function in Figure 3.17 is quite close to the one given by the *modularization* in Figure 3.6. And the prediction for reality in Figure 3.18 is also quite close. The confidence bands are somewhat wider due to the Cauchy assumption, but are again much closer to those from the modular analysis than those from the full normal analysis.

In fact, we first because aware of this issue, not through the size of the bias, but because the MCMC under the full normal model would not mix well. This is illustrated by looking at the trace plots of σ^2_{170} with each of the three approaches. Clearly there is little mixing under the full normal model, and reasonable mixing under the modular normal and Cauchy models. Indeed, improvement of mixing of the MCMC is a general feature of incorporating modularity.

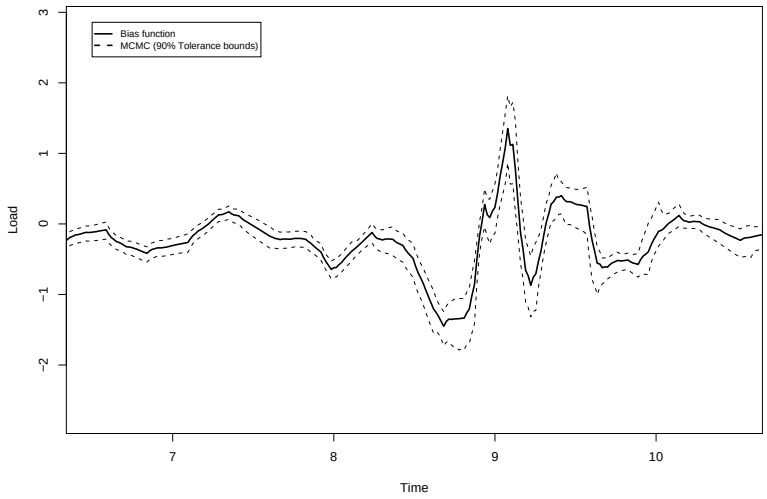

Figure 3.17: Posterior mean of the bias function with 90% confidence bands under the full Bayes approach with Cauchy model assumption.

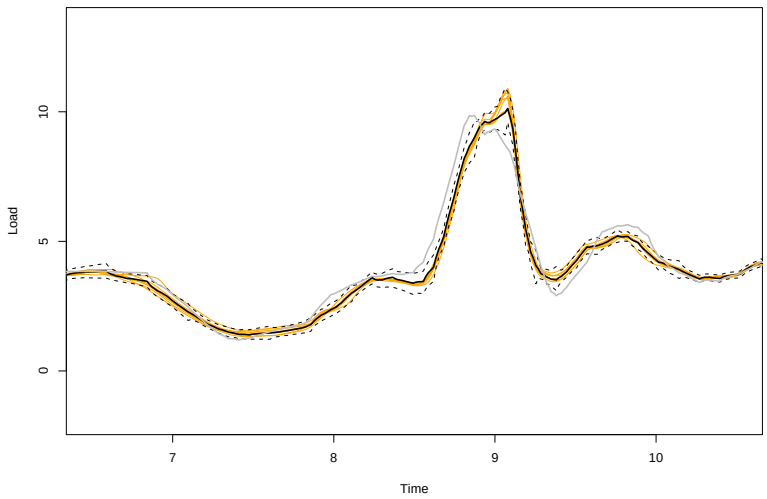

Figure 3.18: Posterior mean of the prediction for reality with 90% confidence bands under the full Bayes with Cauchy model assumption.

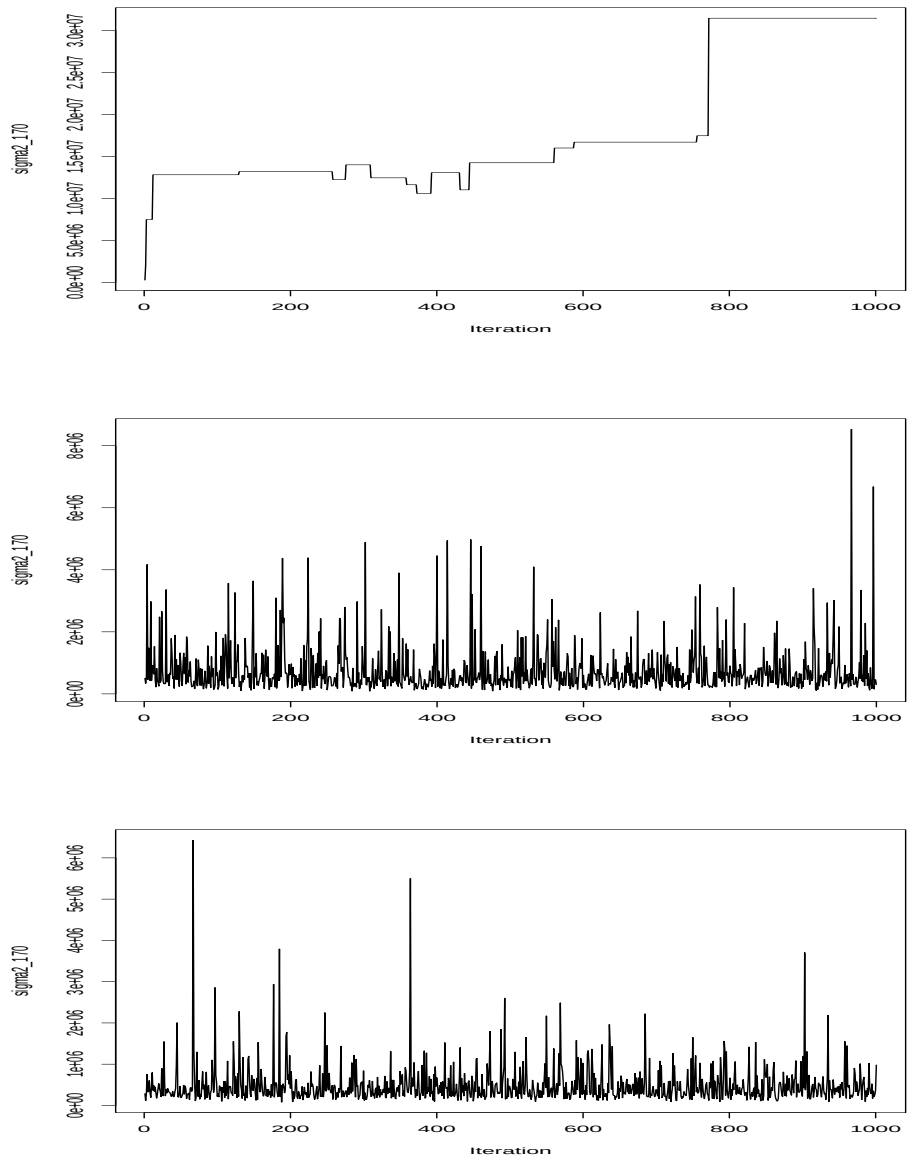

Figure 3.19: Trace plots for σ^2_{170}. Upper: global Bayesian analysis with Normal assumption; Middle: global Bayesian analysis with Cauchy assumption; Lower: *modularized* Bayesian analysis with Normal assumption.

Chapter 4

Multiple Computer Models

Computer model inputs often occur at different scales, which we will call *structural* and *controllable*. In engineering systems, for instance, structural inputs are the basic elements of the computer model, e.g., the finite element representation of the object being studied. Controllable inputs are inputs that can be varied for a given structure.

An example is computer modeling of vehicle suspension systems, as discussed in Chapter 3. The structural inputs here relate to the type of vehicle for which the suspension system is being modeled. The controllable inputs are those that can be varied for a given vehicle, such as damping and vehicle weight. (Note that it is not the whole vehicle that is being modeled, in which case weight would be determined by the structural inputs; it is just the suspension system of the vehicle that is being modeled.)

These are not precise definitions and relate, in part, to operational uses of the computer model. The computer model (and possibly the field data) will typically be run over a wide range of the controllable inputs, but only a modest number of values of the structural inputs, as changing the structural inputs requires very significant alterations in the computer model. Because of this, we will refer to the problem as that of dealing with multiple computer models.

Often, prediction is required across computer models, i.e., for a new set of structural inputs. This was the case in Chapter 3, where field data and computer model runs (for varying controllable inputs) were available for the structural input

labeled Condition A, and prediction was desired for the structural input labeled Condition B (a different vehicle type) based solely on computer model runs (for varying controllable inputs). With no way to quantify the difference between the two structural inputs, we were forced to make predictions under the simplifying assumption that the bias for the two situations was the same.

In this chapter we suppose that we have available data from multiple computer models (multiple structural inputs), and that we have surrogate input variables that relate to the structural input. (We do not deal with the structural input directly, because it is too high-dimensional – e.g., the entire finite element representation of a vehicle suspension system.) For the example of the vehicle suspension systems, the available surrogate variables are the $\boldsymbol{x}_{nom}$ discussed in Chapter 3, which were different for Condition A and Condition B. Assuming we have data from a number of such differing structural inputs, we can include $\boldsymbol{x}_{nom}$ as a variable in the analysis, thus hoping to at least partly account for the differences in the structural inputs. The controllable inputs are then the calibration parameters and the unknown manufacturing variations $\boldsymbol{\delta}$, as discussed in Chapter 3. Note the oddity, for this testbed example, that we do not directly deal with the original inputs, which were given as $\boldsymbol{x} = \boldsymbol{x}_{nom} + \boldsymbol{\delta}$, as would be done if these were for a single computer model; this is not appropriate here because changes in $\boldsymbol{x}_{nom}$ imply that there were serious changes made in the code itself. Hence we separately include $\boldsymbol{x}_{nom}$ and $\boldsymbol{\delta}$ as inputs; implicit is the assumption that the manufacturing variations $\boldsymbol{\delta}$ have the same impact across computer models.

Performing computations across multiple computer models is very expensive, so we develop an alternative approach that greatly simplifies the computation. Specifically, we utilize composite basis elements selected by a principal components

analysis; this greatly reduces the dimension of the problem.

4.1 Introduction

In this chapter, we consider multiple time-dependent computer models. The difference across the models is essentially the nominal inputs for the key characteristics inputs. Thus, we need the nominal inputs $\boldsymbol{x}_{nom}$ to identify which code to run and the manufacturing variation $\boldsymbol{\delta}$ to run that particular code. For simplicity, we use $\boldsymbol{x}$ to represent the nominal inputs in the rest of this chapter. $\boldsymbol{x}$ corresponds to the configuration $\boldsymbol{v}$ in Equation (1.3).

Given the nominal inputs $\boldsymbol{x} = (x_1, \ldots, x_{d_1})$ and the manufacturing variation inputs $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_{d_1})$ to the system, denote the "real" response over time t as $y^R(\boldsymbol{x}, \boldsymbol{\delta}; t)$. Field measurement of the real response has error, so that we write the r^{th} replicate field measurement as

$$y_r^F(\boldsymbol{x}, \boldsymbol{\delta}; t) = y^R(\boldsymbol{x}, \boldsymbol{\delta}; t) + \varepsilon_r(t)\,, \tag{4.1}$$

where the $\varepsilon_r(\cdot)$'s are independent mean zero Gaussian processes as in Chapter 3. Additionally, the actual manufacturing variations $\boldsymbol{\delta}$ are essentially unknown in the field runs; we must also take that into account.

We assume that the computer codes are based on the same physics, so $\boldsymbol{u} = (u_1, \ldots, u_{d_2})$ denotes common calibration/tuning parameters of all computer models. The model output is then of the form $y^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}; t)$.

The connection between model output and reality is then expressed as

$$y^R(\boldsymbol{x}, \boldsymbol{\delta}; t) = y^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}^*; t) + b(\boldsymbol{x}, \boldsymbol{\delta}; t)\,, \tag{4.2}$$

where $\boldsymbol{u}^*$ is the true value of the (vector) calibration parameter; $y^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}^*; t)$

is the model response at time t and the true value of $\boldsymbol{u}$; and $b(\boldsymbol{x}, \boldsymbol{\delta}; t)$ is the associated bias.

We illustrate the analysis of multiple computer codes by the following test bed case study. The problem is about predicting loads from stressful events on vehicle suspension systems (manufacturing platforms) over time. A single platform has been described and studied in Chapter 3. The computer model requires $d = 9$ inputs, with $d_1 = 7$ key characteristics inputs and $d_2 = 2$ calibration parameters. We use $\boldsymbol{u}^* = (u_1^*, u_2^*)$ to represent the true values for the calibration parameters. The other inputs are key characteristics associated with the tested vehicles, with the nominal values for these inputs equal to $\boldsymbol{x}$ and variations equal to $\boldsymbol{\delta}$, i.e., for a particular tested vehicle with specified variation $\boldsymbol{\delta}$, the corresponding inputs are equal to $\boldsymbol{x} + \boldsymbol{\delta}$. Table 3.1 in Chapter 3 gives information about these inputs for the single platform studied there.

Each tested vehicle belongs to a different manufacturing platform. The key characteristics for that tested vehicle are usually unknown, centering around the nominal values $\boldsymbol{x}$ for the corresponding platform. The difference is due to manufacturing variation. We use $\boldsymbol{\delta}$ to represent such variation, and $\boldsymbol{\delta}^*$ stands for the unknown true variation. Our study involves six manufacturing platforms, labeled as A, B, C, D, E, F. Table 4.1 gives the nominal values for these platforms.

Each datum is a time history of load at a site on the vehicle suspension system. The computer model data are available under $m = 6$ platforms, namely A-F. Each platform has $n = 64$ runs at various inputs values. The values of the calibration parameters and manufacturing variations are selected according to a Latin Hypercube Design (LHD), and are common to all the platforms.

The field data consists of the runs from 20 vehicles, 5 from Platform A, 4 from

Platform	A	B	C	D	E	F
x_1	0.5	0.5	0.5	0.5	0.5	0.5
x_2	0.5	0.5	0.5	0.5	0.5	0.5
x_3	0.5	0.5	0.5	0.5	0.5	0.5
x_4	0.5	0.5	0.5	0.5	0.5	0.5
x_5	0.5	2.5	4.5	6.5	8.5	10.5
x_6	0.5	2.5	4.5	6.5	8.5	10.5
x_7	0.5	0.5	0.5	0.5	0.5	0.5

Table 4.1: The nominal values for Platform A-F.

Platform B, 4 from Platform C, 3 from Platform E, and 1 from Platform F. All have replicated runs. We label every vehicle in the study as follows by its platform name and the order it has been tested. For example, the first vehicle in Platform A is labeled as A_1. A_1 and F_1 are the two vehicles involved in the analysis in Chapter 3. For illustrative purposes, we only use field runs under Condition A-D and F. The goal is to combine the data across all platforms to predict under Platform E.

Our focus is on the methodological development for the multiple computer code scenario. Figure 4.1 shows the computer model runs and the first field vehicle runs in each platform. Part of the data is simulated, as described in Section C.1 of Appendix C, since we were unable to obtain sufficient model-run or field data to adequately exercise the methodology.

4.2 Function Representations

Let $\boldsymbol{\delta}_{ij}^*$ be the true manufacturing variation of the j^{th} tested specimen with the nominal value $\boldsymbol{x}_i$. We represent the r^{th} replicate of that tested specimen by $y_r^F\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*; t\right)$. The approach in Chapter 3 uses wavelets to represent these functions, and reduces the number of nonzero coefficients by a hard thresholding al-

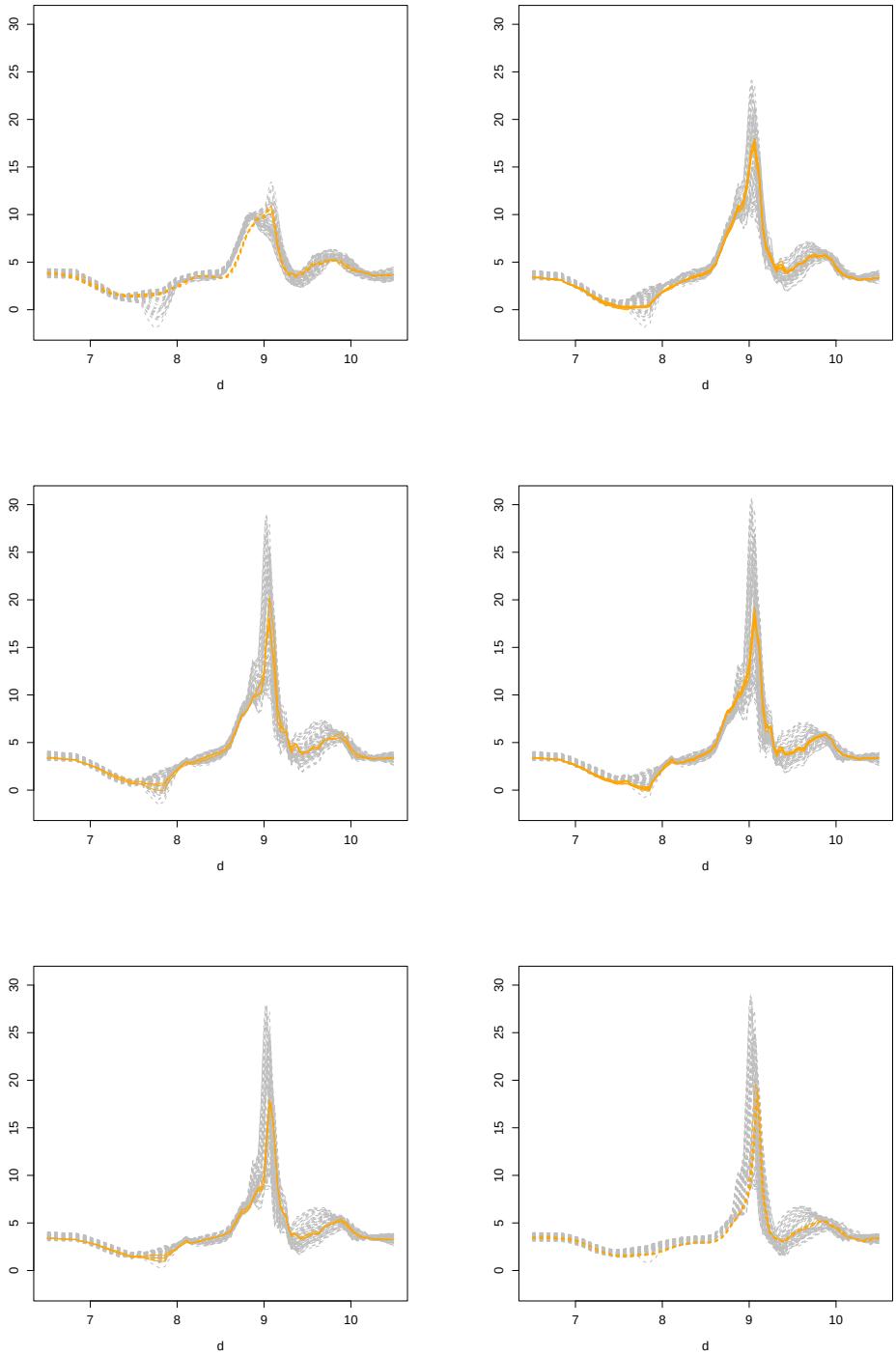

Figure 4.1: The computer model runs (grey) and the first field vehicle runs (orange) in platform A to F.

gorithm (Vidakovic, 1999). This leads to

$$y^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}; t) = \sum_{i \in I} \omega_i^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}) \psi_i(t)$$

$$y_r^F(\boldsymbol{x}, \boldsymbol{\delta}; t) = \sum_{i \in I} \omega_{ir}^F(\boldsymbol{x}, \boldsymbol{\delta}) \psi_i(t)\,, \tag{4.3}$$

where $\{\psi_i(t)\}$ are the wavelet basis elements and I is the index set of nonzero wavelet coefficients after thresholding.

The reduction of the computational expense via the hard thresholding method, however, may not be sufficient when statistical analysis requires many evaluations of the functions $\{\omega_i^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})\}$, or correspondingly the emulators. Consequently, a representation is needed which has fewer coefficients. We achieve this by using functional principle components (Ramsay and Silverman, 1997), as follows.

Consider a set of functions $\{y_i(t), i = 1, \ldots, N\}$. Define the covariance function $v(s,t)$ as

$$v(s,t) = N^{-1} \sum_{i=1}^{N} y_i(s) y_i(t)\,.$$

Functional principle component analysis seeks a set of eigen-functions $\{\xi_i(t)\}$, which satisfy

$$\int v(t,s)\xi_i(s)ds =< v(t,\cdot), \xi_i >= \rho_i \xi_i(t)\,. \tag{4.4}$$

In Equation (4.4), $< \cdot, \cdot >$ is the inner product defined on the functional space, and $\{\rho_i\}$ are the eigen values.

Suppose the functions are originally written in terms of the basis expansion

$$y_i(t) = \sum_{k=1}^{K} \omega_{ik} \psi_k(t)\,,$$

where K is the number of basis elements being used, $\{\psi_k(t), k = 1, \ldots, K\}$ are the basis elements and $\{\omega_{ik}\}$ are the coefficients. To compute the functional principle components, we first discretize the functions to an equally spaced fine grid $\{t_1, \ldots, t_n\}$. This yields

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{\Psi}\,,$$

where $\boldsymbol{Y}$ is an $N \times n$ matrix, $(\boldsymbol{Y})_{ij} = y_i(t_j)$; $\boldsymbol{W}$ is an $N \times K$ matrix, $(\boldsymbol{W})_{ij} = \omega_{ij}$; and $\boldsymbol{\Psi}$ is a $k \times n$ matrix, $(\boldsymbol{\Psi})_{ij} = \psi_i(t_j)$. The sample covariance matrix is thus

$$v(t, s) = N^{-1}\psi(t)'\boldsymbol{W}'\boldsymbol{W}\psi(s)\,. \tag{4.5}$$

Letting $\xi_i(t) = \sum_{k=1}^{K} b_{ik}\psi_k(t) = \boldsymbol{\psi}(t)'\boldsymbol{b}_i$, combining Equation (4.4) and Equation (4.5) yields

$$N^{-1}\boldsymbol{W}'\boldsymbol{W}\left(\int \boldsymbol{\Psi}(t)\boldsymbol{\Psi}(t)'d_t\right)\boldsymbol{b}_i = \rho_i\boldsymbol{b}_i\,.$$

For an orthonormal basis such as wavelets,

$$\int \boldsymbol{\Psi}(t)\boldsymbol{\Psi}(t)'d_t = \boldsymbol{I}\,.$$

Therefore, we can obtain the eigen functions $\{\xi_i(t)\}$ by solving the eigen equations

$$N^{-1}\boldsymbol{W}'\boldsymbol{W}\boldsymbol{b}_i = \rho_i\boldsymbol{b}_i\,. \tag{4.6}$$

Letting $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$, we have

$$[\psi_1(t), \ldots, \psi_k(t)] = [\xi_1(t), \ldots, \xi_k(t)]\boldsymbol{B}^{-1}\,.$$

As a result,

$$\begin{aligned} y_i(t) &= \sum_{k=1}^{k} \omega_{ik}\psi_k(t) \\ &= (\omega_{i1}, \ldots, \omega_{ik})(\boldsymbol{B}^{-1})^t \begin{pmatrix} \xi_1(t) \\ \vdots \\ \xi_k(t) \end{pmatrix} \\ &= \sum_{k=1}^{K} a_{ik}\xi_k(t)\,, \end{aligned}$$

where $\{a_{ik}, k = 1, \ldots, K\}$ are the eigen coefficients for representing $y_i(t)$ by the set of eigen functions $\{\xi_k(t)\}$.

For the multiple vehicle suspension system data, using the first $p = 10$ principle components captures 99% of the variation in the data. Using these functional principle components yields the following representation:

$$y^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}; t) = \sum_{i=1}^{p} a_i^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})\xi_i(t)\,,$$

$$y_r^F(\boldsymbol{x}, \boldsymbol{\delta}; t) = \sum_{i=1}^{p} a_{ir}^F(\boldsymbol{x}, \boldsymbol{\delta})\xi_i(t)\,. \tag{4.7}$$

The left panel of Figure 4.2 shows the eigen-function with the largest eigen-value. This function alone can capture the functional shape close to the two potholes - the most important feature of the data. To show the accuracy of the functional principle component representation, we show the reconstructed curve of the first field run of vehicle A_1 in the red curve in the right panel of Figure 4.2, together with the corresponding original data that has been plotted as black.

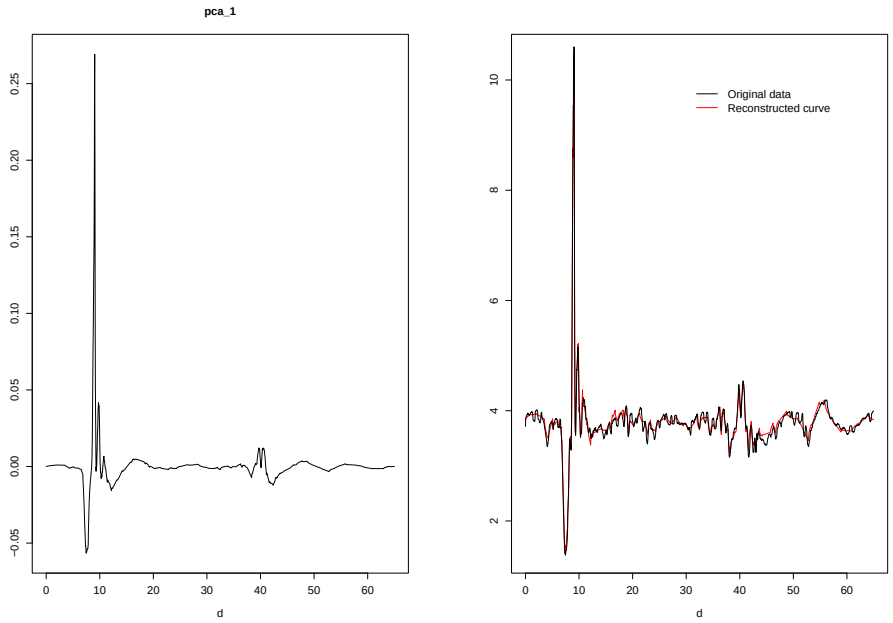

Figure 4.2: Left: the first principle component. Right: the original data of the first field run (black) and the reconstructed functional curve using the first ten principle components (red).

4.3 The Validation Model

4.3.1 Notation

We use the following notation throughout the rest of this chapter.

m:	number of studied computer codes; $m = 6$ in the example.
n:	number of runs of each computer code; $n = 64$ in the example.
p:	number of eigen coefficients being used; $p = 10$ in the example.
i:	code type; $i = 1$ stands for code A, $i = 6$ for code F in the example.
n_i:	number of tested specimen for code type i.
r_{ij}:	number of replicates for the j^{th} tested specimen of the i^{th} type.
$\boldsymbol{x}_i$:	nominal inputs of the i^{th} type.
$\boldsymbol{\delta}^*_{ij}$:	true manufacturing variations of the j^{th} tested specimen of the i^{th} type.
$\boldsymbol{z}$:	$\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{\delta})$.

4.3.2 Formulation

The *SAVE* methodology (Bayarri *et al.*, 2005a) formulates the computer model validation problem as "reality = model + bias". In Chapter 3, we applied this formulation to the wavelet coefficients. In this chapter, we apply the formulation to the eigen-coefficients. For the j^{th} vehicle within the i^{th} platform, denoting the k'th eigen-coefficient for reality and bias, respectively, by $a_k^R(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$ and $a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$, we have

$$a_k^R(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) = a_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), \quad \forall k\,. \tag{4.8}$$

For complex computer systems, the $a_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})$'s, as functions of $(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})$, can only be observed at limited design points. Therefore emulators are needed and will be considered in the next section.

The field data, $a_{kr}^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$, are viewed as reality, with measurement error,

$$a_{kr}^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}) = a_k^R(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}) + \epsilon_{ijkr}^F, \quad \forall k\,. \tag{4.9}$$

We assume that the field measurement errors, ϵ_{ijkr}^F in Equation (4.9), are normally distributed with mean zero and are independent across replications r and across the coefficient index k. This is similar to the wavelet approach in Chapter 3, where there is discussion of this assumption.

4.3.3 Building the emulator

Treating the nominal inputs as separate inputs to the computer model, we build an emulating system for the k^{th} eigen coefficient, $a_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})$, by assigning a Gaussian stochastic process prior distribution

$$a_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}) \sim \mathrm{GP}\left(\mu_k, \sigma_k^{2M}\mathbb{C}\mathrm{orr}_k^{M_1}(\cdot,\cdot)\mathbb{C}\mathrm{orr}_k^{M_2}(\cdot,\cdot)\right). \tag{4.10}$$

In Equation 4.10, the $\mathbb{C}\text{orr}_k^{M_1}$'s are across-code correlations and the $\mathbb{C}\text{orr}_k^{M_2}$ are within-code correlations. We use the power exponential family to model these correlations as

$$\mathbb{C}\text{orr}_k^{M_1}\left(\boldsymbol{x}, \boldsymbol{x}^{'}\right) = \exp\left(-\sum_{l=1}^{d_1} \beta_{kl}^{x} \mid \boldsymbol{x}_l - \boldsymbol{x}_l^{'} \mid^{\alpha_{kl}^{x}}\right) \tag{4.11}$$

and

$$\mathbb{C}\text{orr}_k^{M_2}\left((\boldsymbol{\delta}, \boldsymbol{u}), (\boldsymbol{\delta}^{'}, \boldsymbol{u}^{'})\right) = \exp\left(-\sum_{l=1}^{d_1} \beta_{kl}^{\delta} \mid \boldsymbol{\delta}_l - \boldsymbol{\delta}_l^{'} \mid^{\alpha_{kl}^{\delta}} - \sum_{l=1}^{d_2} \beta_{kl}^{u} \mid \boldsymbol{u}_l - \boldsymbol{u}_l^{'} \mid^{\alpha_{kl}^{u}}\right). \tag{4.12}$$

We assume that the design points, $\boldsymbol{z} = (\boldsymbol{\delta}, \boldsymbol{u})$, are chosen to be the same across the multiple codes. Defining

$$\left(\Sigma_k^{M_1}\right)_{ij} = \mathbb{C}\text{orr}_k^{M_1}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right), \qquad \left(\Sigma_k^{M_2}\right)_{ij} = \mathbb{C}\text{orr}_k^{M_2}\left(\boldsymbol{z}_i, \boldsymbol{z}_j\right),$$

the likelihood can then be written as

$$\boldsymbol{a}_k^M \sim \mathrm{N}\left(\mu_k^M \mathbf{1}, \sigma_k^{2M} \Sigma_k^{M_1} \otimes \Sigma_k^{M_2}\right), \tag{4.13}$$

where $\boldsymbol{a}_k^M$ is the vector of the k^{th} coefficients from the computer model runs,

$$(\boldsymbol{a}_k^M)_{n(i-1)+j} = a_k^M(\boldsymbol{z}_i, \boldsymbol{z}_j) \quad (i = 1, \ldots, m; j = 1, \ldots, n).$$

Model-run data is typically so limited as to require some constraints on the GP parameters. For the k^{th} coefficient, define the distance between $\boldsymbol{\delta}$ and $\boldsymbol{\delta}^{'}$ as,

$$d_k^{\delta}(\boldsymbol{\delta}, \boldsymbol{\delta}^{'}) = \sum_{l=1}^{d_1} \beta_{kl}^{\delta} \mid \boldsymbol{\delta}_l - \boldsymbol{\delta}_l^{'} \mid^{\alpha_{kl}^{\delta}}.$$

We extend this distance to the distance defined in the nominal space, $d_k^x(\boldsymbol{x}, \boldsymbol{x}')$, by an adjustment factor ϕ_k ($\phi_k < 1.0$),

$$d_k^x(\boldsymbol{x}, \boldsymbol{x}') = \phi_k d_k^\delta(\boldsymbol{x}, \boldsymbol{x}')\,.$$

We then define the GP parameters as

$$\beta_{kl}^x = \phi_k \beta_{kl}^\delta, \qquad \text{and} \qquad \alpha_{kl}^x = \alpha_{kl}^\delta\,.$$

We thus are assuming common roughness parameters across the computer models, but allow range parameters to differ by a multiplicative constant across computer models. This keeps the total number of unknown parameters reasonable, while allowing some variation across the models.

4.3.4 Modeling the bias

We model the k^{th} bias coefficient $a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$ for the j^{th} tested vehicle within the i^{th} platform by

$$a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) = b_k(\boldsymbol{x}_i) + \epsilon_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)\,, \tag{4.14}$$

where $b_k(\boldsymbol{x})$ is the mean bias of the k^{th} coefficient for the field specimen with nominal input $\boldsymbol{x}$ and $\epsilon_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$ is the difference between the total bias and the mean bias $b_k(\boldsymbol{x}_i)$; this is due to the differing $\boldsymbol{\delta}_{ij}^*$. We assign $b_k(\boldsymbol{x}_i)$ a Gaussian stochastic process prior

$$b_k^B(\cdot) \sim \text{GP}\left(\mu_k^B, \sigma_k^{2B}\mathbb{C}\text{orr}_k^B(\cdot,\cdot)\right)\,, \tag{4.15}$$

with power exponential correlation function specified as

$$\mathbb{C}\text{orr}_k^B(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\sum_{l=1}^{d_1} \beta_{kl}^B \mid \boldsymbol{x}_l - \boldsymbol{x}_l' \mid^{\alpha_{kl}^B}\right)\,.$$

The other term, $\epsilon_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$, is modeled as a normal distribution $\sim \mathrm{N}(0, \sigma_k^{2B^\epsilon})$. This is an approximation, necessitated by the fact that the $\boldsymbol{\delta}_{ij}^*$ are unknown, and hence there is no information in the data about the $\epsilon_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$. Modeling these differences as random error is similar to assuming a nugget in *GaSP* if there are small unknown systematic errors.

4.3.5 Other prior specification

The context specific priors for $\boldsymbol{u}^*, \boldsymbol{\delta}^*$ are chosen as follows:

$$\begin{aligned}
&\pi(u_1) = \pi(u_2) = \text{ Uniform on } [0.125, 0.875]\,,\\
&\pi(\delta_1) = \pi(\delta_2) \sim N(0, 0.1111^2) \text{ truncated to}[-0.3333, 0.3333]\,,\\
&\pi(\delta_3) \sim N(0, 0.09723^2) \text{ truncated to } [-0.2917, 0.2917]\,,\\
&\pi(\delta_4) = \pi(\delta_7) \sim N(0, 0.1026^2) \text{ truncated to}[-0.3077, 0.3077]\,,\\
&\pi(\delta_5) \sim N(0, 0.04903^2) \text{ truncated to}[-0.1471, 0.1471]\,,\\
&\pi(\delta_6) \sim N(0, 0.1176^2) \text{ truncated to}[-0.3529, 0.3529]\,.
\end{aligned}$$

These are the same as in Chapter 3. Since there are replicates for the field runs, we can use non-informative priors for μ_k^B, σ_k^{2B}, $\sigma_k^{2B^\epsilon}$ and σ_k^{2F} as follows:

$$\pi(\mu_k^B) \propto 1\,, \qquad \pi(\sigma_k^{2B}) \propto \frac{1}{\sigma_k^{2B}}\,,$$

$$\pi(\sigma_k^{2B^\epsilon}) \propto \frac{1}{\sigma_k^{2B^\epsilon}}\,, \qquad \pi(\sigma_k^{2F}) \propto \frac{1}{\sigma_k^{2F}}\,.$$

4.4 Estimation and Analysis

The spirit of "modularity" is to stabilize the MCMC computation, by separating components of the model to the extent possible, using methods such as maximum

likelihood estimates (Bayarri *et al.*, 2005a). Here we separate the analysis into three components: (1) estimating the GP parameters of the emulators; (2) estimating the GP parameters of the biases; (3) Bayesian analysis for the remaining parameters.

4.4.1 Estimation

Let $\boldsymbol{\theta}_k^M$ be the GP parameters for the emulator,

$$\boldsymbol{\theta}_k^M = \left(\mu_k^M, \sigma_k^{2M}, \phi_k^M, \{\alpha_{kl}^\delta, \beta_{kl}^\delta, l = 1, \ldots, d_1\}, \{\alpha_{kl}^u, \beta_{kl}^u, l = 1, \ldots, d_2\}\right) ,$$

and $\widehat{\boldsymbol{\theta}}_k^M$ be the estimates of $\boldsymbol{\theta}$. We use iterative weighted maximum likelihood to determine $\widehat{\boldsymbol{\theta}}_k^M$. The algorithm is detailed in Section C.2 of Appendix C. These parameters are fixed for the rest of the analysis.

We can then write the predictive distribution for $a_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})$ as

$$a_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}) \mid \widehat{\boldsymbol{\theta}}_k^M, \boldsymbol{a}_k^M \sim \mathrm{N}\left(\widehat{m}_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}), \widehat{V}_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})\right) , \tag{4.16}$$

where $\widehat{m}_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})$ and $\widehat{V}_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})$ are

$$\widehat{\mu}_k^M + \left(\widehat{\gamma}_k^{M_1}(\boldsymbol{x})^t \left(\widehat{\boldsymbol{\Sigma}}_k^{M_1}\right)^{-1}\right) \otimes \left(\widehat{\gamma}_k^{M_2}(\boldsymbol{\delta}, \boldsymbol{u})^t \left(\widehat{\boldsymbol{\Sigma}}_k^{M_2}\right)^{-1}\right) (\boldsymbol{a}_k^M - \widehat{\mu}_k^M \mathbf{1}) ,$$

$$\widehat{\sigma}_k^{2M}\left(\mathbf{1} - \widehat{\gamma}_k^{M_1}(\boldsymbol{x})^t \left(\widehat{\boldsymbol{\Sigma}}_k^{M_1}\right)^{-1} \widehat{\gamma}_k^{M_1}(\boldsymbol{x}) \widehat{\gamma}_k^{M_2}(\boldsymbol{\delta}, \boldsymbol{u})^t \left(\widehat{\boldsymbol{\Sigma}}_k^{M_2}\right)^{-1} \widehat{\gamma}_k^{M_2}(\boldsymbol{\delta}, \boldsymbol{u})\right) .$$

In the above equations, $\mathbf{1}$ represents the vector of ones, $\widehat{\boldsymbol{\Sigma}}_k^{M_1}$ is an $m \times m$ correlation matrix and $\widehat{\boldsymbol{\Sigma}}_k^{M_2}$ is an $n \times n$ correlation matrix obtained by plugging $\widehat{\boldsymbol{\theta}}_k^M$ into Equation (4.13). $\widehat{\gamma}_k^{M_1}(\boldsymbol{x})$ and $\widehat{\gamma}_k^{M_2}(\boldsymbol{x})$ are defined as

$$\widehat{\gamma}_k^{M_1}(\boldsymbol{x}) = \left(\widehat{\mathbb{C}}\mathrm{orr}_k^{M_1}(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, \widehat{\mathbb{C}}\mathrm{orr}_k^{M_1}(\boldsymbol{x}_m, \boldsymbol{x})\right)'$$

$$\widehat{\gamma}_k^{M_2}(\boldsymbol{x}) = \left(\widehat{\mathbb{C}}\mathrm{orr}_k^{M_2}((\boldsymbol{\delta}_1, \boldsymbol{u}_1), (\boldsymbol{\delta}, \boldsymbol{u})), \ldots, \widehat{\mathbb{C}}\mathrm{orr}_k^{M_2}((\boldsymbol{\delta}_n, \boldsymbol{u}_n), (\boldsymbol{\delta}, \boldsymbol{u}))\right)' ,$$

where $\widehat{\mathbb{C}}\text{orr}_k^{M_1}$ and $\widehat{\mathbb{C}}\text{orr}_k^{M_2}$ are the estimated correlation functions (plug in $\widehat{\boldsymbol{\theta}}_k^M$). Equation (4.16) is the emulator we use for $a_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u})$.

We estimate the GP parameters in the biases as follows. Combining Equation (4.8), Equation (4.9), Equation (4.14), and Equation (4.16), we have,

$$a_{kr}^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) = \widehat{m}_k^M\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*\right) + a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) + \epsilon_{kr}^F + \mathrm{N}(0, \widehat{V}_k^M\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*\right)).$$

We further approximate it by

$$a_{kr}^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) \approx \widehat{m}_k^M(\boldsymbol{x}_i, \widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}}) + a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) + \epsilon_{kr}^F + \mathrm{N}(0, \widehat{V}_k^M\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*\right)),$$

where $\widehat{\boldsymbol{\delta}}$ and $\widehat{\boldsymbol{u}}$ are the prior means for $\boldsymbol{\delta}, \boldsymbol{u}$ ($\widehat{\boldsymbol{\delta}} = \mathbf{0}$, and $\widehat{\boldsymbol{u}} = \mathbf{0}.5$ for the vehicle suspension systems) and $\widehat{m}_k^M(\boldsymbol{x}_i, \widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}})$ is defined in Equation (4.16). Letting $\bar{a}_k^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) = \frac{1}{n_{ij}} \sum_{r=1}^{n_{ij}} a_{kr}^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$, it follows that the distribution of $\bar{a}_k^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) - \widehat{m}_k^M(\boldsymbol{x}_i, \widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}})$ is close to a normal distribution, with mean equal to $a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$ and variance equal to $\sigma_k^{2F}/r_{ij} + \widehat{V}_k^M\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*\right)$. $\widehat{V}_k^M\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*\right)$ is usually small, since we are interpolating the space of $\boldsymbol{\delta}^*, \boldsymbol{u}^*$ for a given $\boldsymbol{x}_i$. Consequently, we can obtain estimates

$$\widehat{\boldsymbol{\theta}}_k^B = (\widehat{\alpha}_{kl}^B, \widehat{\beta}_{kl}^B) \quad (l = 1, \ldots, p_1)$$

by applying Welch's GASP code (with a nugget that accommodates the normal errors above) to the $\{\bar{a}_k^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) - \widehat{m}_k^M(\boldsymbol{x}_i, \widehat{\boldsymbol{\delta}}, \widehat{\boldsymbol{u}})\}$. We fix $\boldsymbol{\theta}_k^B$ at $\widehat{\boldsymbol{\theta}}_k^B$ for the rest of the analysis. Thus we use plug-in estimates of the correlation matrices, $\widehat{\Sigma}_k^B$, for the $b_k^B(\boldsymbol{x})$'s. However, we do not fix μ_k^B and σ_k^{2B}; we include them in the subsequent Bayesian analysis.

4.4.2 The posterior distributions

For $i = 1 \ldots m$, $j = 1 \ldots n_i$, $k = 1 \ldots p$, $r = 1 \ldots r_{ij}$, we rewrite Equation (4.8), Equation (4.9) and Equation (4.14) as

$$\begin{aligned} a_{kr}^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) &= a_k^R(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) + \epsilon_{ijkr}^F \,, \\ a_k^R(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) &= a_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) \,, \\ a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) &= b_k(\boldsymbol{x}_i) + \epsilon_{ijk}^B \,. \end{aligned} \tag{4.17}$$

In Equation (4.17), the ϵ_{ijkr}^F's are independent, $\epsilon_{ijkr}^F \sim \mathrm{N}(0, \sigma_k^{2F})$; the ϵ_{ijk}^B's are independent, $\epsilon_{ijk}^B \sim \mathrm{N}(0, \sigma_k^{2B^\epsilon})$; $a_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) \sim \mathrm{N}\left(\widehat{m}_k^M\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*\right), \widehat{V}_k^M\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*\right)\right)$ is given in Equation (4.16); and $b_k(\cdot)$ is a Gaussian process with the correlation parameters estimated as in Section 4.4.1.

Summarizing the field data by the sufficient statistics $\bar{a}_{ijk}^F = \frac{1}{r_{ij}} \sum_{r=1}^{r_{ij}} a_{kr}^F(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$ and $s_{ijk}^2 = \sum_{r=1}^{r_{ij}} \left(a_{kr}^F\left(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*\right) - \bar{a}_{ijk}^F\right)^2$, we have

$$\bar{a}_{ijk}^F \sim \mathrm{N}\left(a_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), \frac{1}{r_{ij}}\sigma_k^{2F}\right) ,$$

and

$$\frac{1}{\sigma_k^{2F}} \sum_{i=1}^{m} \sum_{j=1}^{n_i} s_{ijk}^2 \sim \chi^2_{\sum_{i,j}(r_{ij}-1)} \,.$$

The posterior distributions can be written as the product of the following two factors,

$$\prod_k \prod_{i,j} \pi\left(a_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) \mid a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), b_k(\boldsymbol{x}_i), \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*, \mu_k^B, \sigma_k^{2F}, \sigma_k^{2B}, \sigma_k^{2B^\epsilon}, \mathbf{D}\right) , \tag{4.18}$$

and

$$\prod_k \prod_{i,j} \pi\left(a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), b_k(\boldsymbol{x}_i), \delta_{ij}^*, \boldsymbol{u}^*, \mu_k^B, \sigma_k^{2F}, \sigma_k^{2B}, \sigma_k^{2B^\epsilon} \mid \mathbf{D}\right) . \tag{4.19}$$

The full conditional distributions for the $a_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$'s in Equation (4.18) are independent normal distributions, with mean

$$\frac{\widehat{V}_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*)\left(\bar{a}_{ijk}^F - a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)\right)}{\widehat{V}_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + \sigma_k^{2F}/r_{ij}} + \frac{\sigma_k^{2F}/r_{ij}\left(\widehat{m}_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*)\right)}{\widehat{V}_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + \sigma_k^{2F}/r_{ij}},$$

and variance given by

$$\frac{\widehat{V}_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*)\sigma_k^{2F}/r_{ij}}{\widehat{V}_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + \sigma_k^{2F}/r_{ij}}.$$

We define the integrated likelihood $\mathrm{L}^I\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \{\mu_k^B\}, \{\sigma_k^{2F}\}, \{\sigma_k^{2B}\}, \{\sigma_k^{2B^\epsilon}\}\right)$ by integrating out $a_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*)$, $a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)$ and $a_k^B(\boldsymbol{x}_i)$ with respect to their prior distributions. We represent the integrated likelihood as

$$\mathrm{L}^I\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \{\mu_k^B\}, \{\sigma_k^{2F}\}, \{\sigma_k^{2B}\}, \{\sigma_k^{2B^\epsilon}\}\right) = \prod_k \mathrm{L}_k^I\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \mu_k^B, \sigma_k^{2F}, \sigma_k^{2B}, \sigma_k^{2B^\epsilon}\right),$$

with

$$\mathrm{L}_k^I\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \mu_k^B, \sigma_k^{2F}, \sigma_k^{2B}, \sigma_k^{2B^\epsilon} \mid \mathbf{D}\right) \propto \left(\frac{1}{\sigma_k^{2F}}\right)^{\frac{1}{2}\sum_{i,j}(r_{ij}-1)} \frac{1}{\mid \mathbf{E} + \sigma_k^{2B}\widehat{\Sigma}_k^B \mid^{1/2}}$$

$$\exp\left(-\frac{1}{2\sigma_k^{2F}}\sum_{i,j} s_{ijk}^2 - \frac{1}{2}\left(\bar{\boldsymbol{a}}_k^F - \widehat{\boldsymbol{m}}_k^M - \mu_k^B\mathbf{1}\right)^t\left(\mathbf{E} + \sigma_k^{2B}\widehat{\Sigma}_k^B\right)^{-1}\left(\bar{\boldsymbol{a}}_k^F - \widehat{\boldsymbol{m}}_k^M - \mu_k^B\mathbf{1}\right)\right). \tag{4.20}$$

where the diagonal matrix $\mathbf{E}$ is

$$\mathbf{E} = \mathrm{diag}\left(\widehat{V}_k^M(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + \sigma_k^{2B^\epsilon} + \frac{1}{r_{ij}}\sigma_k^{2F}\right).$$

The draws from the distribution in Equation (4.19) are obtained from an MCMC algorithm. The algorithm is detailed in Section 4.5. Here, we derive the conditional distributions that are needed for the algorithm.

- $\pi\left(b_k(\boldsymbol{x}_i), i = 1, \ldots, m \mid \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*, \mu^B_k, \sigma^{2F}_k, \sigma^{2B}_k, \sigma^{2B^\epsilon}_k, \mathbf{D}\right)$:

  Letting $\boldsymbol{b}_k = (b_k(\boldsymbol{x}_1), \ldots, b_k(\boldsymbol{x}_m))'$, we obtain the conditional distribution for $\boldsymbol{b}_k$ by integrating out the $a^B_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij})$'s in Equation (4.19). Letting

$$\bar{\boldsymbol{a}}^F_k = \left(\bar{a}^F_{ijk}, i = 1, \ldots, m; j = 1, \ldots n_i\right)' ,$$

  and

$$\widehat{\boldsymbol{m}}^M_k = \left(\widehat{m}^M_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*), i = 1, \ldots, m; j = 1, \ldots n_i\right)' ,$$

  $\boldsymbol{b}_k$ follows a multivariate normal distribution, with mean vector

$$\left(\bar{\boldsymbol{a}}^F_k - \widehat{\boldsymbol{m}}^M_k\right) - \mathbf{E}\left(\mathbf{E} + \sigma^{2B}_k \widehat{\Sigma}^B_k\right)^{-1}\left(\bar{\boldsymbol{a}}^F_k - \widehat{\boldsymbol{m}}^M_k - \mu^B_k \mathbf{1}\right) ,$$

  and covariance matrix

$$\mathbf{E} - \mathbf{E}\left(\mathbf{E} + \sigma^{2B}_k \widehat{\Sigma}^B_k\right)^{-1}\mathbf{E} .$$

- $\pi\left(a^B_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}) \mid b_k(\boldsymbol{x}_i), \delta^*_{ij}, \boldsymbol{u}^*, \mu^B_k, \sigma^{2F}_k, \sigma^{2B}_k, \sigma^{2B^\epsilon}_k, \mathbf{D}\right)$:

  Conditional on $b_k(\boldsymbol{x}_i), i = 1, \ldots, m$, $a^B_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij})$ are independent normal variables with mean

$$\frac{\sigma^{2B^\epsilon}_k}{\sigma^{2B^\epsilon}_k + \sigma^{2F}_k / r_{ij} + \widehat{V}^M_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)}\left(\bar{a}^F_{ijk} - \widehat{m}^M_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)\right)$$

$$+ \frac{\sigma^{2F}_k / r_{ij} + \widehat{V}^M_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)}{\sigma^{2B^\epsilon}_k + \sigma^{2F}_k / r_{ij} + \widehat{V}^M_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)} a^B_k(\boldsymbol{x}_i) ,$$

  and variance

$$\frac{\sigma^{2B^\epsilon}_k\left(\sigma^{2F}_k / r_{ij} + \widehat{V}^M_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)\right)}{\sigma^{2B^\epsilon}_k + \sigma^{2F}_k / r_{ij} + \widehat{V}^M_k(\boldsymbol{x}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)} .$$

- $\left(\mu_k^B \mid \{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \sigma_k^{2F}, \sigma_k^{2B}, \sigma_k^{2B^\epsilon}, \mathbf{D}\right)$:

  Further integrating out $b_k(\boldsymbol{x}_i)$ in the likelihood leads to the conditional distribution for μ_k^B as normal with mean and variance given by

$$\left(\mathbf{1}'(\mathbf{E}+\sigma_k^{2B}\widehat{\Sigma}_k^B)^{-1}\mathbf{1}\right)^{-1}\left(\mathbf{1}'(\mathbf{E}+\sigma_k^{2B}\widehat{\Sigma}_k^B)^{-1}(\bar{\boldsymbol{a}}_k^F-\widehat{\boldsymbol{m}}_k^M)\right),$$

  and

$$\left(\mathbf{1}'(\mathbf{E}+\sigma_k^{2B}\widehat{\Sigma}_k^B)^{-1}\mathbf{1}\right)^{-1}.$$

- $\left(\sigma_k^{2B} \mid b_k(\boldsymbol{x}_i), a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), \delta_{ij}^*, \boldsymbol{u}^*, \mu_k^B, \sigma_k^{2F}, \mathbf{D}\right)$:

  The conditional distribution for σ_k^{2B} is

$$\sigma_k^{2B} \sim \text{IG}\left(\frac{1}{2}\text{rank}\left(\widehat{\Sigma}_k^B\right), \frac{1}{2}\left(\boldsymbol{b}_k-\mu_k^B\mathbf{1}\right)^t\left(\widehat{\Sigma}_k^B\right)^-\left(\boldsymbol{b}_k-\mu_k^B\mathbf{1}\right)\right). \tag{4.21}$$

- $\left(\sigma_k^{2B^\epsilon} \mid b_k(\boldsymbol{x}_i), a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), \delta_{ij}^*, \boldsymbol{u}^*, \mu_k^B, \sigma_k^{2F}, \mathbf{D}\right)$:

  The conditional distribution for $\sigma_k^{2B^\epsilon}$ is

$$\sigma_k^{2B^\epsilon} \sim \text{IG}\left(\frac{1}{2}\sum_{i,j} r_{ij}, \frac{1}{2}\sum_{i,j}\left(a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)-b_k(\boldsymbol{x}_i)\right)^2\right). \tag{4.22}$$

4.5 The MCMC Algorithm

Grouping the parameters as $\{a_k^B(\boldsymbol{x}_i), a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)\}$, $\{\delta_{ij}^*, \boldsymbol{u}^*, \mu_k^B, \sigma_k^{2F}\}$, and $\{\sigma_k^{2B}, \sigma_k^{2B^\epsilon}\}$, we use Gibbs sampling to draw the samples from their joint posterior distribution. The MCMC algorithm has the following seven steps at each iteration.

For iteration h,

Step 1: Draw $\mu_k^{B(h)}$ from

$$\pi\left(\mu_k^B \mid \{\boldsymbol{\delta}_{ij}^{*(h-1)}\}, \boldsymbol{u}^{*(h-1)}, \sigma_k^{2F(h-1)}, \sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)}, \mathbf{D}\right) .$$

The distribution has been given in Section 4.4.2.

Step 2: Given $\mu_k^{B(h)}$, $\sigma_k^{2F(h-1)}$, $\sigma_k^{2B(h-1)}$ and $\sigma_k^{2B^\epsilon(h-1)}$, draw $\{\boldsymbol{\delta}_{ij}^{*(h)}\}, \boldsymbol{u}^{*(h)}$ from

$$\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^* \mid \{\mu_k^{B(h)}, \sigma_k^{2F(h-1)}, \sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)}\}, \mathbf{D}\right) ,$$

by a Metropolis-Hastings algorithm described later. The Metropolis-Hastings algorithm usually yields highly correlated iterations, so we cycle through this step for 200 times (fixing the other parameters) and keep only the last sample of the parameters.

Step 3: Given $\mu_k^{B(h)}$, $\sigma_k^{2B(h-1)}$, $\sigma_k^{2B^\epsilon(h-1)}$, $\{\boldsymbol{\delta}_{ij}^{*(h)}\}$, and $\boldsymbol{u}^{*(h)}$, draw $\{\sigma_k^{2F(h)}\}$ from

$$\left(\sigma_k^{2F} \mid \{\boldsymbol{\delta}_{ij}^{*(h)}\}, \boldsymbol{u}^{*(h)}, \{\mu_k^{B(h)}, \sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)}\}, \mathbf{D}\right) ,$$

by a Metropolis-Hastings algorithm described later. Again, we cycle through this step for 200 times and keep the last draw.

Step 4: Given $\{\mu_k^{B(h)}\}$, $\{\boldsymbol{\delta}_{ij}^{*(h)}\}, \boldsymbol{u}^{*(h)}$, $\{\sigma_k^{2F(h)}\}$ and $\{\sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)}\}$, draw $b_k^{(h)}(\boldsymbol{x}_i)$ from

$$\pi\left(b_k(\boldsymbol{x}_i), i = 1, \ldots, m \mid \boldsymbol{\delta}_{ij}^{*(h)}, \boldsymbol{u}^{*(h)}, \mu_k^{B(h)}, \sigma_k^{2F(h)}, \sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)}, \mathbf{D}\right) .$$

The distribution has been given in Section 4.4.2.

Step 5: Given $b_k^{(h)}(\boldsymbol{x}_i)$, $\{\mu_k^{B(h)}\}$, $\{\boldsymbol{\delta}_{ij}^{*(h)}\}, \boldsymbol{u}^{*(h)}$, $\{\sigma_k^{2F(h)}\}$ and $\{\sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)}\}$, draw $\{a_k^{B(h)}(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*)\}$ from

$$\pi\left(a_k^B(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*) \mid b_k^{(h)}(\boldsymbol{x}_i), \delta_{ij}^{*(h)}, \boldsymbol{u}^{*(h)}, \mu_k^{B(h)}, \sigma_k^{2F(h)}, \sigma_k^{2B(h)}, \sigma_k^{2B^\epsilon(h)}, \mathbf{D}\right) .$$

The distribution has been given in Section 4.4.2.

Step 6: Given $b_k^{(h)}(\boldsymbol{x}_i), a_k^{B(h)}(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^{*(h)}), \mu_k^{B(h)}$, draw samples $\{\sigma_k^{2B(h)}\}$ and $\{\sigma_k^{2B^\epsilon(h)}\}$ according to Equation (4.21) and Equation (4.22).

Step 7: Given $a_k^{B(h)}(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^{*(h)}), a_k^{B(h)}(\boldsymbol{x}_i), \boldsymbol{\delta}_{ij}^{*(h)}, \boldsymbol{u}^{*(h)}, \mu_k^{B(h)}, \sigma_k^{2F(h)}, \sigma_k^{2B(h)}, \sigma_k^{2B^\epsilon(h)}$, draw samples of $a_k^{M(h)}(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^{*(h)}, \boldsymbol{u}^{*(h)})$ according to distribution (4.18).

The Metropolis-Hastings algorithm in **Step 2** propose new samples of the $\boldsymbol{\delta}_{ij}^{*h}$'s and $\boldsymbol{u}^{*h}$ from mixture distributions of the prior and a local random walk. Let $T_l^\delta = [a_l^\delta, A_l^\delta]$ and $T_l^u = [a_l^u, A_l^u]$ denote the intervals on which the prior densities for the corresponding variables are nonzero, and define

$$T_{ijl}^{*\delta h} = \left[\max(a_l^\delta, \delta_{ijl}^h - 0.01), \min(A_l^\delta, \delta_{ijl}^h + 0.01)\right],$$
$$T_l^{*uh} = \left[\max(a_l^u, u_l^h - 0.01), \min(A_l^u, u_l^h + 0.01)\right].$$

Propose $\{\boldsymbol{\delta}_{ij}^*\}$ and $\boldsymbol{u}^*$ from

$$g(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^* \mid \{\boldsymbol{\delta}_{ij}^{*h}\}, \boldsymbol{u}^{*h}) =$$
$$\prod_{i,j}\prod_{l=1}^{p_1}\left(\frac{9}{10}U(\delta_{ijl}^* \mid T_{ijl}^{*\delta h}) + \frac{1}{10}\pi(\delta_{ijl}^*)\right) \times \prod_{l=1}^{p_2}\left(\frac{1}{2}U(u_l^* \mid T_l^{*uh}) + \frac{1}{2}U(a_l^u, A_1^u)\right).$$

The Metropolis-Hastings Ratio for this proposal distribution is

$$\rho = \frac{\pi\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \{\mu_k^{B(h)}\}, \{\sigma_k^{2F(h-1)}\} \mid \{\sigma_k^{2B(h-1)}\}, \{\sigma_k^{2B^\epsilon(h-1)}\}, \mathbf{D}\right)}{\pi\left(\{\boldsymbol{\delta}_{ij}^{*(h-1)}\}, \boldsymbol{u}^{*(h-1)}, \{\mu_k^{B(h)}\}, \{\sigma_k^{2F(h-1)}\} \mid \{\sigma_k^{2B(h-1)}\}, \{\sigma_k^{2B^\epsilon(h-1)}\}, \mathbf{D}\right)}$$
$$\times \frac{g\left(\{\delta_{ij}^{*(h-1)}\}, \boldsymbol{u}_{ij}^{*(h-1)} \mid \{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*\right)}{g\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^* \mid \{\boldsymbol{\delta}_{ij}^{*(h-1)}\}, \boldsymbol{u}^{*(h-1)}\right)},$$

where

$$\pi\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \{\mu_k^B, \sigma_k^{2F}\} \mid \{\sigma_k^{2B}, \sigma_k^{2B^\epsilon}\}, \mathbf{D}\right) \propto$$

$$\mathrm{L}^I\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \{\mu_k^B, \sigma_k^{2F}, \sigma_k^{2B}, \sigma_k^{2B^\epsilon}\} \mid \mathbf{D}\right) \pi\left(\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*, \{\mu_k^B, \sigma_k^{2F}\}\right) . \qquad (4.23)$$

Set $(\{\boldsymbol{\delta}_{ij}^{*(h)}\}, \boldsymbol{u}^{*(h)}) = (\{\boldsymbol{\delta}_{ij}^*\}, \boldsymbol{u}^*)$ with probability $\min(1, \rho)$, and equal to $(\{\boldsymbol{\delta}_{ij}^{*(h-1)}\}, \boldsymbol{u}^{*(h-1)})$ otherwise.

In **Step 3**, we propose σ_k^{2F} from

$$g_k(\sigma_k^{2F}) = \mathrm{IG}\left(\frac{1}{2}\sum_{i,j}(r_{ij} - 1), \frac{1}{2}\sum_{i,j} s_{ijk}^2\right) ,$$

and calculate the Metropolis-Hastings Ratio

$$\begin{aligned} \rho_k &= \frac{\mathrm{L}_k^I\left(\{\boldsymbol{\delta}_{ij}^{*(h)}\}, \boldsymbol{u}^{*(h)}, \mu_k^{B(h)}, \sigma_k^{2F}, \sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)} \mid \mathbf{D}\right)}{\mathrm{L}_k^I\left(\{\boldsymbol{\delta}_{ij}^{*(h)}\}, \boldsymbol{u}^{*(h)}, \mu_k^{B(h)}, \sigma_k^{2F(h-1)}, \sigma_k^{2B(h-1)}, \sigma_k^{2B^\epsilon(h-1)} \mid \mathbf{D}\right)} \\ &\times \frac{\pi(\sigma_k^{2F})}{\pi(\sigma_k^{2F(h-1)})} \frac{g_k(\sigma_k^{2F(h-1)})}{g_k(\sigma_k^{2F})} . \end{aligned}$$

Set $\sigma_k^{2F(h)} = \sigma_k^{2F}$ with probability $\min(1, \rho_k)$, and equal to $\sigma^{2F(h-1)}$ otherwise.

At the end of MCMC, we have a sample of 1000 draws $(h = 1, \ldots, 1000)$,

$$\left\{\{\boldsymbol{\delta}_{ij}^{*h}\}, \boldsymbol{u}^{*h}, \{a_k^{M(h)}(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), \{a_k^{B(h)}(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^*), b_k^h(\boldsymbol{x}_i), \mu_k^{B(h)}, \sigma_k^{2F(h)}, \sigma_k^{2B(h)}, \sigma_k^{2B^\epsilon(h)}\}\right\} .$$

Our inferences are based on these samples.

4.6 Results

In this section, we illustrate the analysis by the example introduced earlier involving multiple vehicle suspension systems. Figure 4.3 gives posterior histograms for

$\boldsymbol{u}^*$ and for $\boldsymbol{\delta}^*$ for the first tested vehicle in Platform A. Note that these results are reasonably consistent with the corresponding posteriors in Figure 3.5, for the situation in which the first tested vehicle in Platform A was analyzed in isolation.

Posterior distributions of the bias and reality curves are obtained by reconstructing the posterior eigen-coefficients with the functional principle components. For example, the posterior distribution of b with nominal input $\boldsymbol{x}_i$ and manufacturing variations $\boldsymbol{\delta}^*_{ij}$, is represented by the sample curves

$$b^h_{ij}(t) = \sum_{k \in I} a^{bh}_k(\boldsymbol{x}_i, \boldsymbol{\delta}^{*h}_{ij}) \xi_k(t); \quad h = 1, \ldots, 1000\,.$$

The posterior mean curve, $\widehat{b}_{ij}(t)$, is estimated as

$$\widehat{b}_{ij}(t) = \frac{1}{1000} \sum_{h=1}^{1000} b^h(\boldsymbol{x}_i, \boldsymbol{\delta}^{*h}_{ij}; t)\,,$$

with the upper and lower uncertainty (tolerance) bounds at each t, produced by taking the lower $\alpha/2$ and upper $1 - \alpha/2$ quantiles of the posterior distribution of $b_j(\boldsymbol{x}_i; t)$,

$$L^b_{ij}(t) = \frac{\alpha}{2} \text{ quantile of } \{b^h_{ij}(t); h = 1, \ldots, 1000\}$$

$$U^b_{ij}(t) = \left(1 - \frac{\alpha}{2}\right) \text{ quantile of } \{b^h_{ij}(t); h = 1, \ldots, 1000\}\,.$$

Figure 4.4 shows, on the upper panel, the bias function for the first tested vehicle in Platform A and, on the lower panel, the bias function for the first tested vehicle in Platform F. (Recall that these were the two data sets from real vehicles.) It is apparent in Figure 3.6 that the bias function is significantly different from 0 especially in the neighborhood of 8.7 and 9.1 – the location of the first pothole.

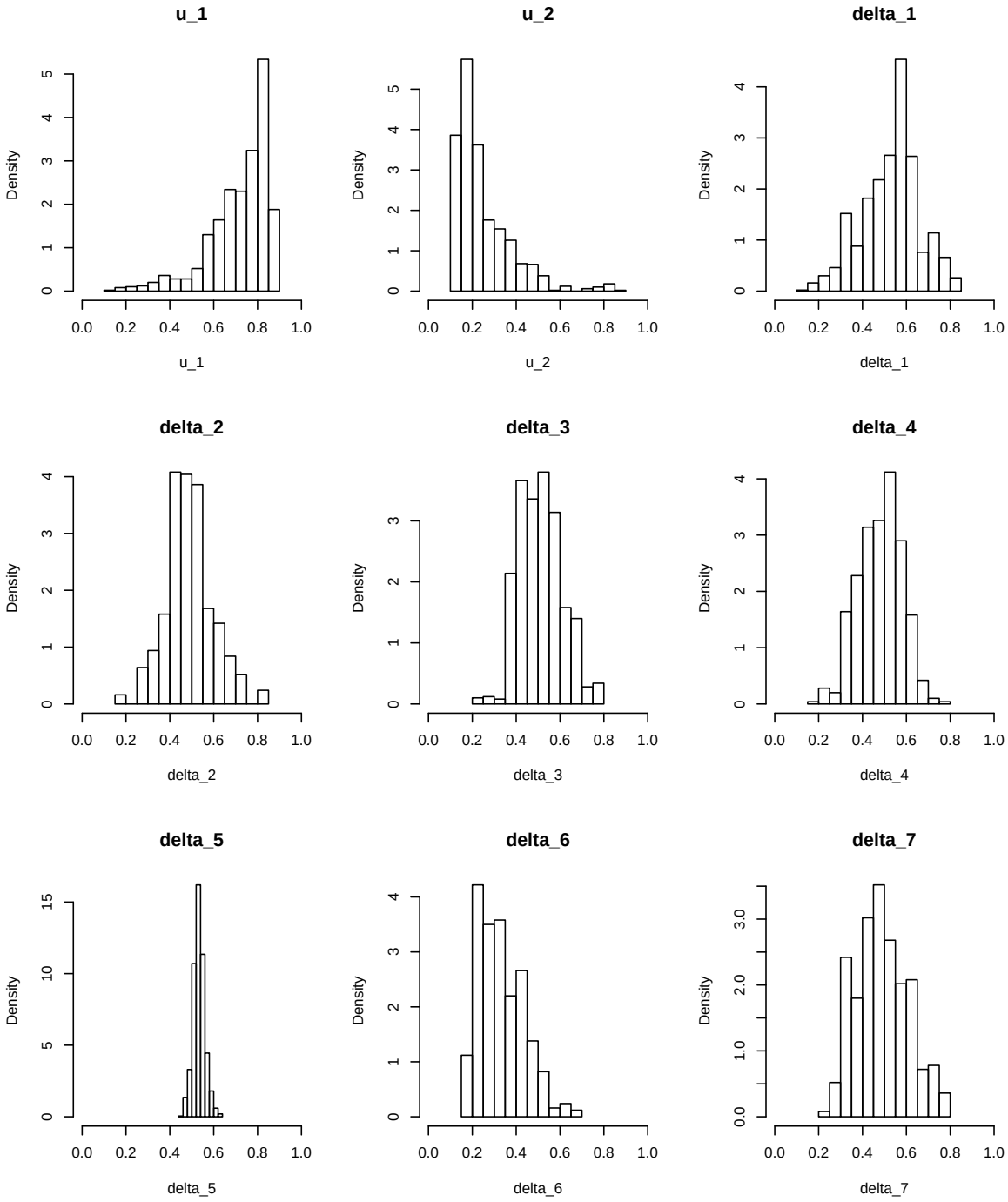

Figure 4.3: Histogram of the posterior draws for the calibration parameters and the manufacturing variations for the first tested vehicle of Platform A.

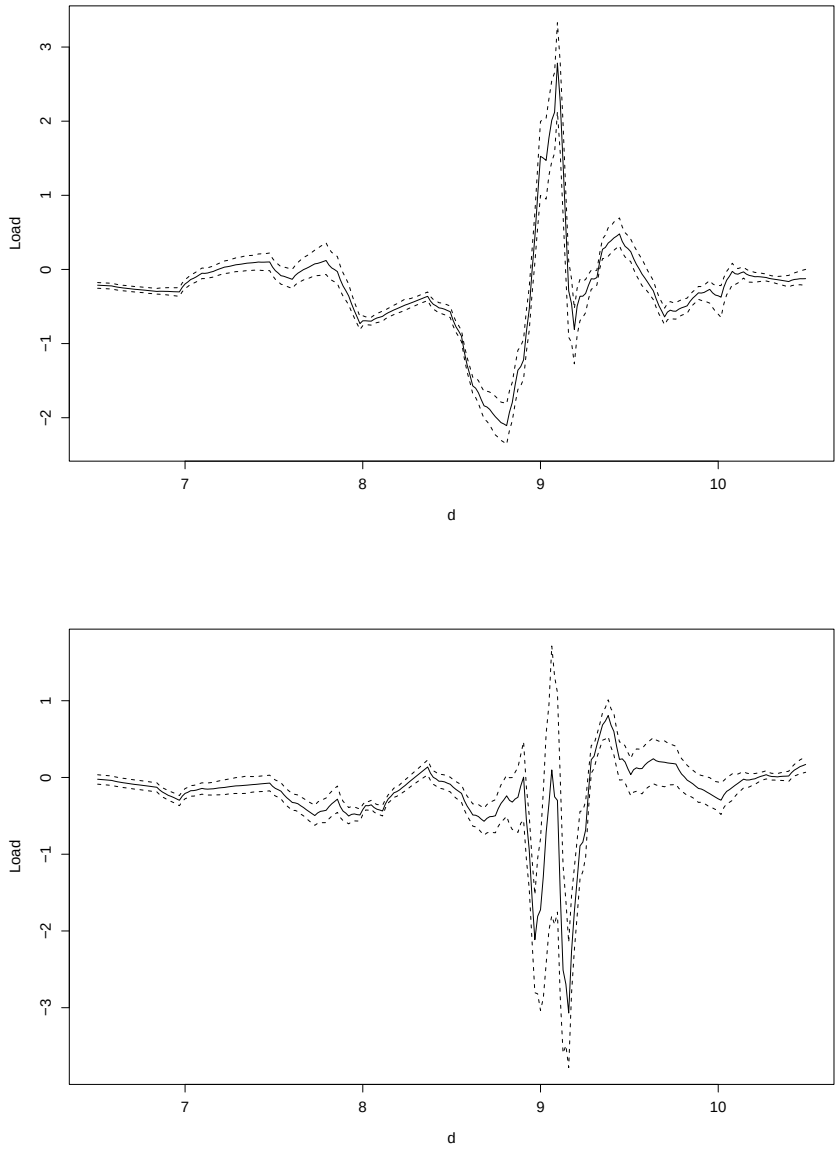

Figure 4.4: Posterior mean curves (in solid lines) with 90% uncertainty bounds (in dashed lines) for the bias function of Vehicle A_1 (in the left panel) and of Vehicle F_1 (in the right panel).

We estimate reality with uncertainty bounds similarly: take the sample of eigen-coefficients $a_k^{Rh}(\boldsymbol{x}, \boldsymbol{\delta}^{*h}) = a_k^{Mh}(\boldsymbol{x}, \boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}) + a_k^{bh}(\boldsymbol{x}, \boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h})$ and form

$$y_{ij}^{Rh}(t) = \sum_k a_k^{Rh}(\boldsymbol{x}_i, \boldsymbol{\delta}_{ij}^{*h}, \boldsymbol{u}^{*h})\xi_k(t), \quad \widehat{y}_{ij}^{R}(t) = \frac{1}{1000}\sum_h y_{ij}^{Rh}(t)\,,$$

$$L_{ij}^{R}(t) = \frac{\alpha}{2} \text{ quantile of } \{y_{ij}^{Rh}(t); h = 1, \ldots, 1000\}\,,$$

$$U_{ij}^{R}(t) = \left(1 - \frac{\alpha}{2}\right) \text{ quantile of } \{y_{ij}^{Rh}(t); h = 1, \ldots, 1000\}\,.$$

Figure 4.5 shows the prediction of reality and associated uncertainty bands for Vehicles A_1 and F_1. These predictions are capturing the major features of the data, although they do not match perfectly. The mismatches seem to be primarily due to the errors resulting from keeping only 10 eigen-basis elements to represent the functions.

Finally, we extrapolate the analysis of Platform A-D, F to Platform E. First, we extrapolate the biases to the new platform. Denote the bias for Platform E as $a_k^B(\boldsymbol{x}_5, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*)$, $a_k^B(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*) \sim \mathrm{N}(b_k(\boldsymbol{x}_5), \sigma_k^{2B^\epsilon})$. To obtain the posterior draws for $b_k(\boldsymbol{x}_5)$, it suffices to draw $\boldsymbol{b}_k^h(\boldsymbol{x}_5)$ at each iteration h from $\pi\left(b_k(\boldsymbol{x}_5) \mid \boldsymbol{b}_k^h, \boldsymbol{\theta}^{(h)}\right)$, which under the Gaussian process prior, is a normal distribution with mean and covariance given by

$$\mathbb{E}(b_k(\boldsymbol{x}_5)) = \mu_b^h + \boldsymbol{c}^t\widehat{\boldsymbol{\Sigma}}_k^-(\boldsymbol{b}^h - \mu_b^h\mathbf{1}), \quad \mathbb{V}\mathrm{ar}\,(b_k(\boldsymbol{x}_5)) = \sigma_k^{2B(h)}\left(1 - \boldsymbol{c}^t\boldsymbol{\Sigma}_k^{-1}\boldsymbol{c}\right)\,,$$

where $\boldsymbol{c} = \left(\mathbb{C}\mathrm{orr}_k^B(\boldsymbol{x}_1, \boldsymbol{x}_5), \ldots, \mathbb{C}\mathrm{orr}_k^B(\boldsymbol{x}_4, \boldsymbol{x}_5), \mathbb{C}\mathrm{orr}_k^B(\boldsymbol{x}_6, \boldsymbol{x}_5)\right)^t$ is the correlation vector between Platform A-D, F and Platform E, and $\widehat{\boldsymbol{\Sigma}}_k^-$ is the correlation matrix of the mean biases of Platform A-D, F. We henceforth obtain a draw $\boldsymbol{a}_k^{Bh}(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*)$ by sampling from $\mathrm{N}\left(b_k^h(\boldsymbol{x}_5), \sigma_k^{2B^\epsilon}\right)$. Figure 4.6 shows the mean curve with uncertainty bands for the bias function of Platform E. This is the bias

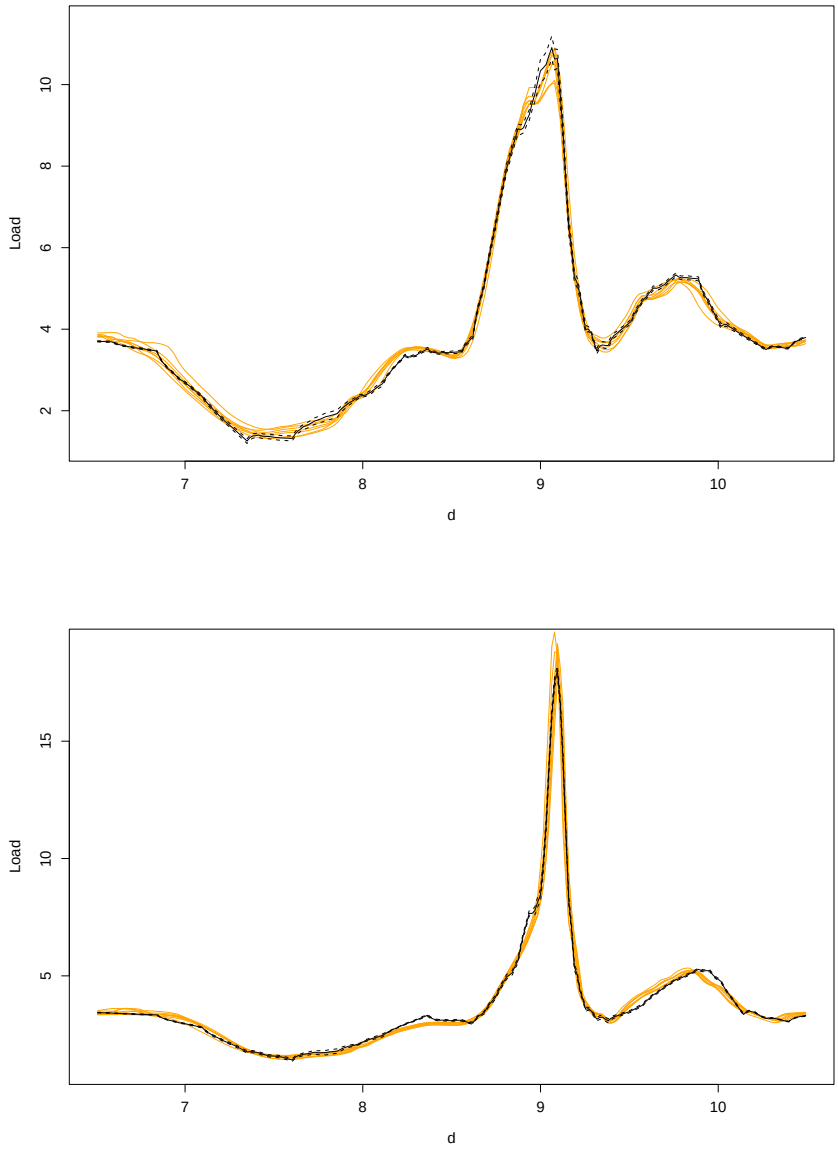

Figure 4.5: Prediction of reality (in solid black lines) with 90% uncertainty bands (in dashed black lines)for Vehicle A_1 (in the upper panel) and F_1 (in the lower panel). The original field data are plotted in orange lines

function for a generic vehicle of type E, thus the uncertainty bands are much wider. Again, as we observe, the bias is significant in the region close to the pothole.

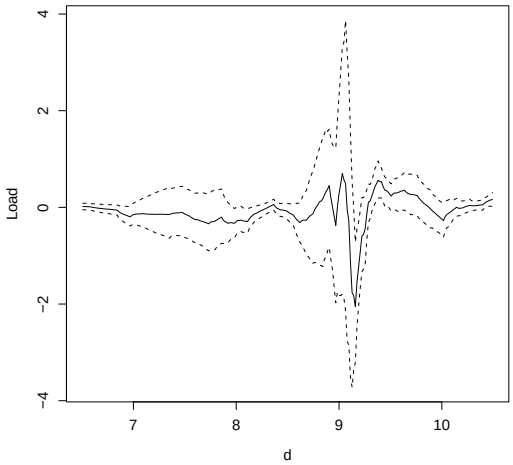

Figure 4.6: Posterior mean curve (in solid lines) with 90% uncertainty bands (in dashed lines) for the bias function of Platform E.

Define the pure model prediction for Platform E as

$$y^M(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*; t) = \sum_k a_k^M \left(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*; t\right) \xi_k(t)\,,$$

where $\boldsymbol{\delta}^* \sim \pi(\boldsymbol{\delta}^*)$ and $\boldsymbol{u}^* \sim \pi(\boldsymbol{u}^* \mid \mathbf{D})$. Consequently, for each h, we obtain the sample of $y^{Mh}(\boldsymbol{x}_5, \boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}; t)$ by drawing $\boldsymbol{\delta}^{*h}$ from the prior and drawing the coefficients from the emulators $N\left(\widehat{m}_k(\boldsymbol{x}_5, \boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h}), \widehat{V}_k(\boldsymbol{x}_5, \boldsymbol{\delta}^{*h}, \boldsymbol{u}^{*h})\right)$. $y^{Rh}(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*; t)$, the bias-corrected prediction of reality for Platform E, is thus obtained by transforming the coefficients of the sum of the bias coefficients and the model coefficients,

$$y^{Rh}(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*; t) = \sum_k \left(a_k^{Mh}\left(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*; t\right) + \boldsymbol{a}_k^{Bh}(\boldsymbol{x}_5, \boldsymbol{\delta}^*, \boldsymbol{u}^*)\right) \xi_k(t)\,.$$

Figure 4.7 shows, on the upper panel, the pure model prediction for a generic vehicle of Platform E and, on the lower panel, the bias-corrected prediction of reality. The real field runs for Platform E are also plotted on the same figures. Note that these data are not included in the analysis. The model prediction mismatches with the real data, while the bias-corrected prediction of reality gives prediction that is much closer to the real data. The uncertainty bands of bias-corrected prediction of reality are also wider.

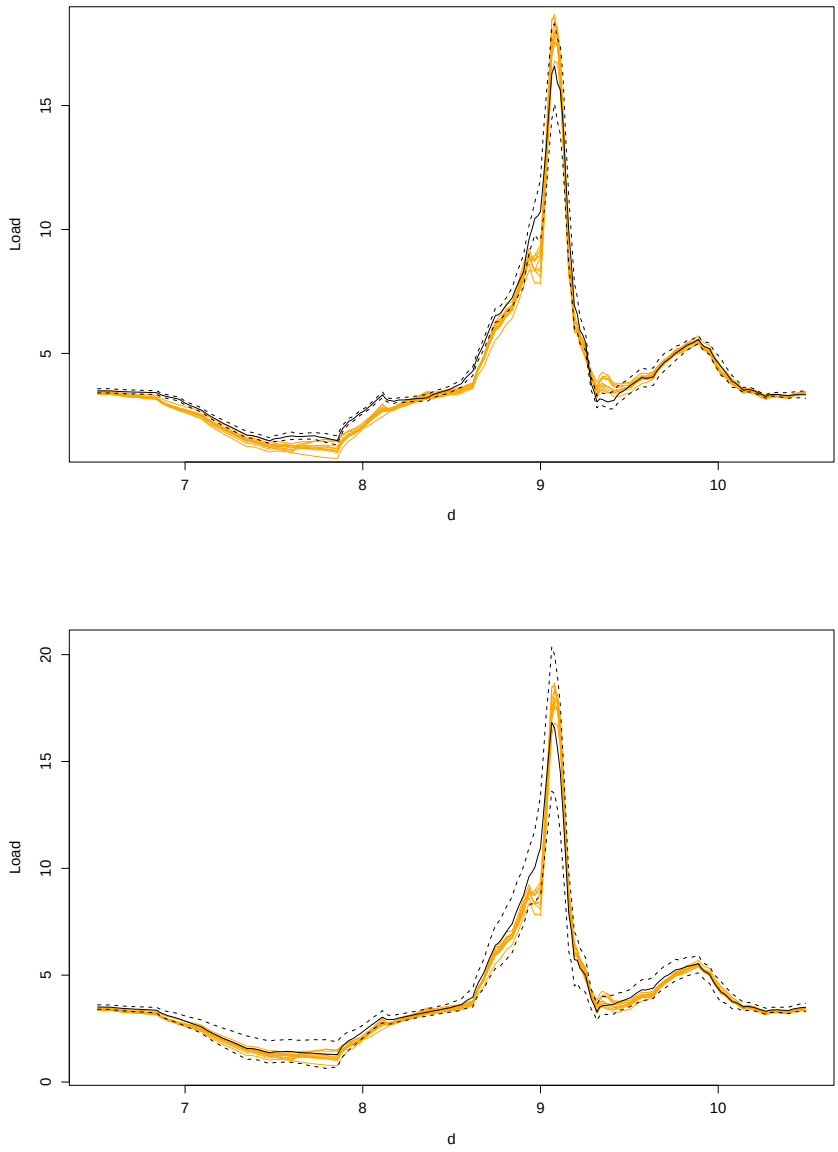

Figure 4.7: Pure model prediction for Platform E (in the upper panel) and Bias-corrected prediction of reality for Platform E (in the lower panel). The posterior mean curves are plotted in solid black lines; the 90% uncertainty bands are plotted in dashed black lines; the original field data are plotted in the orange lines on each of the figures.

Chapter 5

Emulating Computer Models by Dynamic Linear Models

Dynamic linear models (DLMs) are powerful tools for analyzing time series processes. Modeling complex computer systems by DLMs is appealing when the analysis aims at understanding the underlying stochastic structure of the systems. Typical applications arise in environmental science, where data is usually stochastically structured. Many other applications can be found ranging from engineering to biological systems. The input parameters to these systems, in practice, can be time dependent. But for simplicity, we restrict attention to the simple case where they do not vary over time.

The approach considered here for emulating such systems is to combine a Bayesian multivariate dynamic linear model with Gaussian Processes defined on the input parameter space of the computer model. We apply the approach to some simulated data, based on data from a real computer model.

5.1 Bayesian Multivariate Dynamic Linear Model

West and Harrison (1997) gives a comprehensive discussion of Bayesian Dynamic Linear Models. The focus of this chapter is the Multivariate normal DLM, providing a the joint stochastic structure for multiple series. Letting $\boldsymbol{Y}_t = (Y_{t1}, \ldots, Y_{tr})'$ be the vector of observations (obtained at time t, $t = 1, \ldots, T$) on the r studied

series, the DLM is defined by a quadruple,

$$\{\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{V}, \boldsymbol{W}\}_t = \{\boldsymbol{F}_t, \boldsymbol{G}_t, \boldsymbol{V}_t, \boldsymbol{W}_t\} \ .$$

Here, $\boldsymbol{F}_t$ is a $n \times r$ dynamic regression matrix; $\boldsymbol{G}_t$ is a $n \times n$ state evolution matrix; $\boldsymbol{V}_t$ is a $r \times r$ observational variance matrix; and $\boldsymbol{W}_t$ is a $n \times n$ evolution variance matrix. Correspondingly, we can represent the model in terms of the state-space model,

$$\begin{aligned} \text{Observation:} \quad & \boldsymbol{Y}_t = \boldsymbol{F}_t^{'}\boldsymbol{\theta}_t + \boldsymbol{\nu}_t \, , \boldsymbol{\nu}_t \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{V}_t) \, , \\ \text{Evolution:} \quad & \boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t \, , \boldsymbol{\omega}_t \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{W}_t) \, . \end{aligned}$$

In the state-space model, $\boldsymbol{\theta}_t$ is the n-dimensional state vector. The sequence $\{\boldsymbol{\nu}_t\}$ is independent of $\{\boldsymbol{\omega}_t\}$; for $t \neq t^{'}$, $\boldsymbol{\nu}_t$ is independent of $\boldsymbol{\nu}_{t'}$ and $\boldsymbol{\omega}_t$ is independent of $\boldsymbol{\omega}_{t'}$.

Define $D_t = \{\boldsymbol{Y}_t, D_{t-1}\}$, the information set at time t. D_0 is the initial information set. Suppose that the prior distribution for the state vector $\boldsymbol{\theta}_0$ is,

$$\boldsymbol{\theta}_0 \mid D_0 \sim \mathrm{N}(\boldsymbol{m}_0, \boldsymbol{C}_0) \, .$$

West and Harrison (1997) gives the one-step forecast and posterior distributions, for each t, as follows, based on the forward filtering algorithm:

(a) Posterior at $t-1$:

$$(\boldsymbol{\theta}_{t-1} \mid D_{t-1}) \sim \mathrm{N}(\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1}) \, .$$

(b) Prior at t:

$$\boldsymbol{\theta}_t \mid D_{t-1} \sim \mathrm{N}(\boldsymbol{a}_t, \boldsymbol{R}_t)$$

where

$$\boldsymbol{a}_t = \boldsymbol{G}_t\boldsymbol{m}_{t-1} \qquad \text{and} \qquad \boldsymbol{R}_t = \boldsymbol{G}_t\boldsymbol{C}_{t-1}\boldsymbol{G}_t^{'} + \boldsymbol{W}_t \, .$$

(c) One-step forecast:

$$(\boldsymbol{Y}_t \mid D_{t-1}) \sim \mathrm{N}(\boldsymbol{f}_t, \boldsymbol{Q}_t)\,,$$

where

$$\boldsymbol{f}_t = \boldsymbol{F}_t^{'}\boldsymbol{a}_t \qquad \text{and} \qquad \boldsymbol{Q}_t = \boldsymbol{F}_t^{'}\boldsymbol{R}_t\boldsymbol{F}_t + \boldsymbol{V}_t\,.$$

(d) Posterior at t:

$$(\boldsymbol{\theta}_t \mid D_t) \sim \mathrm{N}(\boldsymbol{m}_t, \boldsymbol{C}_t)\,,$$

with

$$\boldsymbol{m}_t = \boldsymbol{a}_t + \boldsymbol{A}_t\boldsymbol{e}_t \qquad \text{and} \qquad \boldsymbol{C}_t = \boldsymbol{R}_t - \boldsymbol{A}_t\boldsymbol{Q}_t\boldsymbol{A}_t^{'}\,,$$

where

$$\boldsymbol{A}_t = \boldsymbol{R}_t\boldsymbol{F}_t\boldsymbol{Q}_t^{-1} \qquad \text{and} \qquad \boldsymbol{e}_t = \boldsymbol{Y}_t - \boldsymbol{f}_t\,.$$

5.2 Modeling Complex Computer Model Outputs

The goal is to develop methodology to model the time series data generated by a complex computer system. Figure 5.1 presents time series data of this type. Each time series in the figure is obtained by running the computer model at an input value $\boldsymbol{z}$, specified to the left of the sequence. Each sequence contains observations at $T = 3000$ consecutive time points. The details of this data are discussed in Appendix D.

Let $y(\boldsymbol{z}, t)$ represent the computer model output at input $\boldsymbol{z}$ and time t ($t = 1, \ldots, T$). Define $\boldsymbol{Y}_{1:t}(\boldsymbol{z}) = (y(\boldsymbol{z}, 1), \ldots, y(\boldsymbol{z}, t))^{'}$. The first six time series (blue colored) in Figure 5.1 are obtained respectively at inputs $\boldsymbol{z}_1 = 0.25, \ldots, \boldsymbol{z}_n = 0.75$. In the following analysis, we use data $\boldsymbol{Y}_{1:T}(\boldsymbol{z}_i)$, $i = 1, \ldots, n$, to build the DLM

model, and predict the time series for an untried $\boldsymbol{z}_0 = 0.5$. The result will be compared with the real computer model output at $\boldsymbol{z}_0$, shown as the red colored sequence in Figure 5.1.

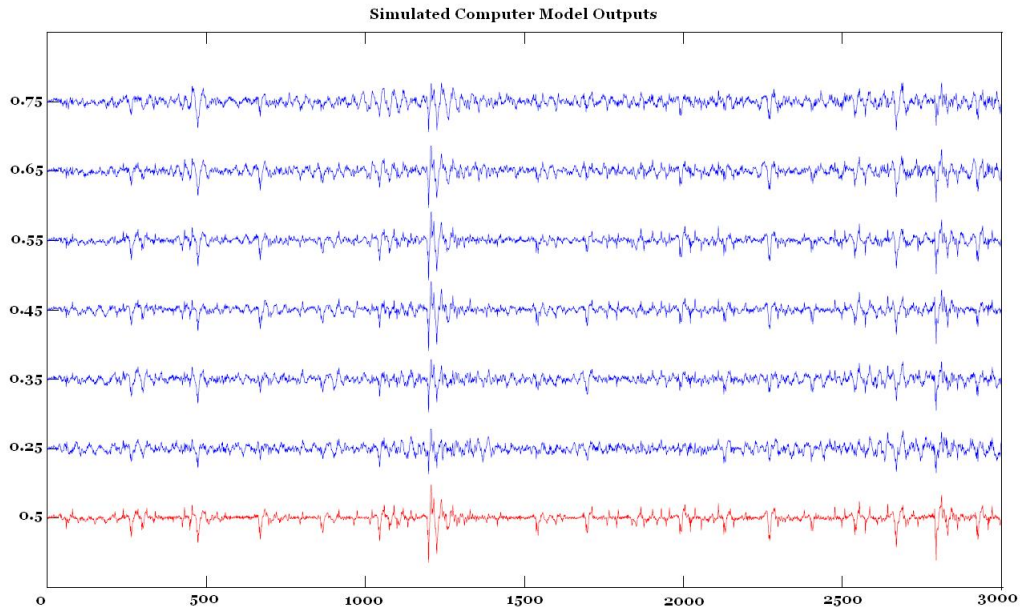

Figure 5.1: The time series outputs of a computer model obtained by simulation at various input values specified at the beginning of each series.

5.2.1 The emulator

The statistical model we will use to emulate the computer model is

$$y\left(\boldsymbol{z},t\right) \;=\; \sum_{j=1}^{p}\phi_{t,j}y(\boldsymbol{z},t-j)+\epsilon_t(\boldsymbol{z})\,, \tag{5.1}$$

where $\epsilon_t(\boldsymbol{z}) \sim \mathrm{GP}(0, v_t\mathbb{C}\mathrm{orr}(\cdot,\cdot))$ with the power exponential correlation defined as $\mathbb{C}\mathrm{orr}(\boldsymbol{z},\boldsymbol{z}^{'}) = \exp(-\sum_i \beta_i \mid \boldsymbol{z}_i - \boldsymbol{z}_i^{'} \mid^{\alpha_i})$. The time-varying autoregressive coefficients, $\boldsymbol{\Phi}_t = (\phi_{t,1},\ldots,\phi_{t,p})^{'}$, are modeled as

$$\boldsymbol{\Phi}_t = \boldsymbol{\Phi}_{t-1} + \boldsymbol{w}_t\,, \qquad \boldsymbol{w}_t \sim \mathrm{N}(0,\boldsymbol{W}_t)\,.$$

$\boldsymbol{\Phi}_0$, v_t, $\boldsymbol{W}_t$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are unknown parameters. We will specify their prior distributions in Section 5.2.2.

Note that according to the above model, the distribution of data observed at $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$, given $\boldsymbol{\Phi}$, is

$$\begin{pmatrix} y(\boldsymbol{z}_1, t) \\ \vdots \\ y(\boldsymbol{z}_n, t) \end{pmatrix} = \begin{pmatrix} y(\boldsymbol{z}_1, t-1) & \ldots & y(\boldsymbol{z}_1, t-p) \\ \vdots & \ddots & \vdots \\ y(\boldsymbol{z}_n, t-1) & \ldots & y(\boldsymbol{z}_n, t-p) \end{pmatrix} \begin{pmatrix} \phi_{t,1} \\ \vdots \\ \phi_{t,p} \end{pmatrix} + \begin{pmatrix} \epsilon_t(\boldsymbol{z}_1) \\ \vdots \\ \epsilon_t(\boldsymbol{z}_n) \end{pmatrix},$$

where $(\epsilon_t(\boldsymbol{z}_1), \ldots, \epsilon_t(\boldsymbol{z}_n))' \sim \mathrm{N}(0, v_t\Sigma)$ with $\Sigma_{i,j} = \mathbb{C}\mathrm{orr}(\boldsymbol{z}_i, \boldsymbol{z}_j)$. This corresponds to a multivariate DLM, concerning the state vector $\{\boldsymbol{\Phi}_t\}$,

$$\{F_t, G_t, \boldsymbol{V}_t, W_t\}_{t=1}^{T},$$

where

$$F_t' = \begin{pmatrix} y(\boldsymbol{z}_1, t-1) & \ldots & y(\boldsymbol{z}_1, t-p) \\ \vdots & \ddots & \vdots \\ y(\boldsymbol{z}_n, t-1) & \ldots & y(\boldsymbol{z}_n, t-p) \end{pmatrix},$$

G_t is the identity matrix of size p, and $\boldsymbol{V}_t = v_t\Sigma$.

The above model can be partly justified as an approximation to the *GaSP* methodology. Let us start with the Gaussian Response Surface Approximation (*GaSP*) methodology for functional data with a time component that was proposed in Bayarri *et al.* (2005b). They put a prior distribution for $y(\boldsymbol{z}, t)$ as

$$y(\boldsymbol{z}, t) \sim \mathrm{GP}\left(\mu, \frac{1}{\lambda}\mathbb{C}\mathrm{orr}(\cdot, \cdot)\right),$$

where $\mathbb{C}\mathrm{orr}(\cdot, \cdot)$ is the correlation function defined on the combined space of $\boldsymbol{z}$ and t. A frequently used correlation function is the separable power exponential

function, which defines the correlation between the random variables $y(\boldsymbol{z}, t)$ and $y(\boldsymbol{z}^{'}, t^{'})$, $\mathbb{C}\text{orr}\left(y(\boldsymbol{z}, t), y(\boldsymbol{z}^{'}, t^{'})\right)$, as

$$\exp\left(-\sum_{i=1}^{d} \beta_i \mid \boldsymbol{z}_i - \boldsymbol{z}_i^{'} \mid^{\alpha_i}\right) \exp\left(-\beta_{(t)} \mid t - t^{'} \mid^{\alpha_{(t)}}\right).$$

In the definition of the correlation function, d is the dimension of the input vector $\boldsymbol{z}$ and $\{\alpha_i, \beta_i, i = 1, \ldots, d\}$ and $\{\alpha_{(t)}, \beta_{(t)}\}$ are hyper-parameters.

For any finite inputs $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ and time series data at those inputs, suppose that $t = 1, \ldots, T$ are the time grid points at which we observe each time series. Let Σ_1 be the $n \times n$ correlation matrix defined on the $\boldsymbol{z}$ space, and Σ_2 be the $T \times T$ correlation matrix on the t space,

$$(\Sigma_1)_{k,l} = \exp\left(-\sum_{i=1}^{d} \beta_i \mid \boldsymbol{z}_{ki} - \boldsymbol{z}_{li}^{'} \mid^{\alpha_i}\right),$$

$$(\Sigma_2)_{k,l} = \exp\left(-\beta_{(t)} \mid t_k - t_l^{'} \mid^{\alpha_{(t)}}\right).$$

We have

$$\begin{pmatrix} \boldsymbol{Y}_{1:T}(\boldsymbol{z}_1) \\ \vdots \\ \boldsymbol{Y}_{1:T}(\boldsymbol{z}_n) \end{pmatrix} \sim \mathrm{N}\left(\mu \times \mathbf{1}, \frac{1}{\lambda}\Sigma_1 \otimes \Sigma_2\right). \tag{5.2}$$

Letting $\boldsymbol{y}_t = \left(y(\boldsymbol{z}_1, t), \ldots, y(\boldsymbol{z}_n, t)\right)^{'}$, we next approximate the model in Equation (5.2) by a DLM.

Let $\mathrm{f}(\boldsymbol{y}_T, \boldsymbol{y}_{T-1}, \ldots, \boldsymbol{y}_1 \mid \boldsymbol{\theta})$ be the joint distribution of $\boldsymbol{y}_T, \boldsymbol{y}_{T-1}, \ldots, \boldsymbol{y}_1$ which is defined in Equation (5.2), where $\boldsymbol{\theta}$ are the hyper-parameters of the model. The

joint distribution can be represented as the product of a sequence of conditionals,

$$\begin{aligned}
&\mathrm{f}(\boldsymbol{y}_T, \boldsymbol{y}_{T-1}, \ldots, \boldsymbol{y}_1 \mid \boldsymbol{\theta}) \\
&\qquad = \left(\prod_{i=p+1}^{T} \mathrm{f}\left(\boldsymbol{y}_i \mid \boldsymbol{y}_{i-1}, \ldots, \boldsymbol{y}_1, \boldsymbol{\theta}\right) \right) \mathrm{f}\left(\boldsymbol{y}_p, \boldsymbol{y}_{p-1}, \ldots, \boldsymbol{y}_1 \mid \boldsymbol{\theta}\right) , \qquad (5.3)
\end{aligned}$$

where p is an integer ranging from 1 to $T-1$. The power exponential correlation is decaying very fast. Suppose after a maximum lag of p time points, that the correlations on t become negligible. Then we have

$$\mathrm{f}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_1, \boldsymbol{\theta}\right) \approx \mathrm{f}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_{t-p}, \boldsymbol{\theta}\right) \qquad \forall t > p\,. \qquad (5.4)$$

The right hand side of Equation (5.4) is a multivariate normal distribution that truncates the original one (given on the left side of that equation) up to step p. Using properties of the Kronecker product given in Appendix A, we have the mean vector of the distribution on the right hand side of Equation (5.4) is

$$\begin{aligned}
\mathbb{E}\left(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, , \ldots, \boldsymbol{y}_{t-p}, \boldsymbol{\theta}\right) &= \left(\boldsymbol{\rho}_{t,t-1:t-p} \otimes \Sigma_1\right)' \left(\tilde{\Sigma}_2 \otimes \Sigma_1\right)^{-1} \begin{pmatrix} \boldsymbol{y}_{t-1} \\ \vdots \\ \boldsymbol{y}_{t-p} \end{pmatrix} \\
&= \left(\boldsymbol{\rho}'_{t,t-1:t-p} \tilde{\Sigma}_2^{-1}\right) \otimes \boldsymbol{I}_{n\times n} \begin{pmatrix} \boldsymbol{y}_{t-1} \\ \vdots \\ \boldsymbol{y}_{t-p} \end{pmatrix} , \qquad (5.5)
\end{aligned}$$

where $\rho(k,l) = \exp\left(-\beta_{(t)} \mid k-l \mid^{\alpha_{(t)}}\right)$, $\boldsymbol{\rho}_{t,t_1:t_2} = (\rho(t,t_1), \ldots, \rho(t,t_2))'$, and $\tilde{\Sigma}_2$ is a $p \times p$ matrix with (k,l) element equal to $\rho(k,l)$. Letting $(\phi_1, \ldots, \phi_p) = \boldsymbol{\rho}'_{t,t-1:t-p}\tilde{\Sigma}_2^{-1}$, Equation (5.5) yields the autoregressive mean vector in Equation (5.1). The covariance matrix of the truncated distribution in Equation (5.4),

$\mathbb{C}\text{ov}(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, \ldots, \boldsymbol{y}_{t-p}, \boldsymbol{\theta})$, is

$$\frac{1}{\lambda^M}\Sigma_1 - \frac{1}{\lambda^M}(\boldsymbol{\rho}'_{t,t-1:t-p} \otimes \Sigma_1)' \tilde{\Sigma}_2^{-1} \otimes \Sigma_1^{-1} \left(\boldsymbol{\rho}'_{t,t-1:t-p} \otimes \Sigma_1\right)$$

$$= \frac{1}{\lambda^M}\left[1 - \left(\boldsymbol{\rho}_{t,t-1:t-p}\right)' \left(\tilde{\Sigma}_2\right)^{-1} \left(\boldsymbol{\rho}_{t,t-1:t-p}\right)\right] \Sigma_1 \,. \quad (5.6)$$

Defining $v_t = \frac{1}{\lambda^M}\left[1 - \left(\boldsymbol{\rho}_{t,t-1:t-p}\right)' \left(\tilde{\Sigma}_2\right)^{-1} \left(\boldsymbol{\rho}_{t,t-1:t-p}\right)\right]$ results in a covariance matrix of the desired form $v_t\Sigma_1$, where Σ_1 does not depend on t.

Combining Equation (5.5) and Equation (5.6) leads to

$$y\left(\boldsymbol{z}, t\right) = \sum_{j=1}^{p} \phi_j y(\boldsymbol{z}, t-j) + \epsilon_t(\boldsymbol{z}) \,, \qquad \epsilon_t(\boldsymbol{z}) \sim \text{GP}(0, v_t \mathbb{C}\text{orr}(\cdot,\cdot)) \,. \quad (5.7)$$

This is generalized to be a non-stationary time series by allowing time-varying autoregressive coefficients and time-varying variances, resulting in the model in Equation (5.1).

5.2.2 The prior distributions

To complete the model specifications, we need to specify the prior distributions for $\{\boldsymbol{\Phi}_0\}$, $\{\boldsymbol{W}_t, t = 1, \ldots, T\}$, $\{v_t, t = 1, \ldots, T\}$ and $\{\alpha_i, \beta_i, i = 1, \ldots, d\}$.

The prior distributions for $\boldsymbol{W}_t$ and v_t are specified sequentially, by employing two discounting factors δ_1 and δ_2,

$$v_t^{-1} \mid \mathrm{D}_{t-1} \sim \text{Gamma}(\delta_1 n_{t-1}/2, \delta_1 d_{t-1}/2) \,, \quad (5.8)$$

and

$$\boldsymbol{W}_t \mid \mathrm{D}_{t-1} = (1-\delta_2) C_{t-1}/\delta_2 \,. \quad (5.9)$$

The values for (n_0, d_0, m_0, C_0) will be pre-specified. $C_{t-1} = \mathbb{C}\text{ov}\left(\Phi_{t-1} \mid \mathrm{D}_{t-1}\right)$ can be calculated by the forward filtering algorithm. $\{n_t\}$ and d_t will be updated sequentially. The updating equation is given in Section 5.3.

For the spatial parameters $\boldsymbol{\alpha} = \{\alpha_i\}$ and $\boldsymbol{\beta} = \{\beta_i\}$, we use the Jeffereys' rule prior $\pi\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right)$ discussed in Berger *et al.* (2001) and Paulo (2005),

$$\pi\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right) \propto \mid I\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right) \mid^{1/2} \propto \sqrt{\mid \mathrm{tr}(\Sigma^{-1}\dot{\Sigma})^2 \mid},$$

where $I\left(\boldsymbol{\alpha}, \boldsymbol{\beta}\right)$ is the Fisher information matrix, and $\dot{\Sigma} = \frac{\partial \Sigma}{\partial(\boldsymbol{\alpha}, \boldsymbol{\beta})}$.

5.3 MCMC

An MCMC method is used to draw joint samples from the posterior distributions $\pi\left(\{v_1, \ldots v_T\}; \{\Phi_1, \ldots, \Phi_T\}; \{\boldsymbol{\alpha}, \boldsymbol{\beta}\} \mid \mathrm{D}_T\right)$. The MCMC algorithm has the following three steps within each iteration.

At iteration i:

Step 1: Sample $\left(\{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\} \mid D_T, \{v_1^{(i-1)}, \ldots, v_T^{(i-1)}\}, \{\boldsymbol{\Phi}_1^{(i-1)}, \ldots, \boldsymbol{\Phi}_T^{(i-1)}\}\right)$ by the Metropolis-Hastings algorithm. We fix α at 2 for the analysis of the example data in Figure 5.1.

Step 2: Sample $\left(\{v_1^{(i)}, \ldots v_T^{(i)}\} \mid D_T, \{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}\right)$. This is complicated and will be described later.

Step 3: Sample $\left(\{\boldsymbol{\Phi}_1^{(i)}, \ldots, \boldsymbol{\Phi}_T^{(i)}\} \mid D_T, \{v_1^{(i)}, \ldots v_T^{(i)}\}, \{\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}\}\right)$.

Given $\{v_t\}$, make the draw $\boldsymbol{\Phi}_T^{(i)}$ from $\left(\boldsymbol{\Phi}_T \mid D_T\right)$. The distribution $\left(\boldsymbol{\Phi}_T \mid D_T\right)$ is known after applying the forward filtering algorithm in Section 5.1,

$$\left(\Phi_T \mid D_T, \{v_1, \ldots, v_T\}\right) \sim \mathrm{MVN}\left(\boldsymbol{m}_T, \boldsymbol{C}_T\right) .$$

Then we make draws of $\Phi_t^{(i)}, t = T-1, \ldots, 1$ recursively from

$$(\Phi_t \mid D_T, \Phi_{t+1}, \{v_1, \ldots, v_T\}) \sim \text{MVN}\left((1-\delta_2)m_t + \delta_2\Phi_{t+1}, (1-\delta_2)C_t\right) .$$

To describe the method to make draws from $(\{v_1, \ldots v_T\} \mid D_T, \{\boldsymbol{\alpha}, \boldsymbol{\beta}\})$, we first give in Theorem 5.1 the forward filtering algorithm for the model defined in Equation (5.1) with discounting updates for $\boldsymbol{W}_t$ and v_t in Equation (5.8) and Equation (5.9).

Theorem 5.1. *Suppose $\boldsymbol{W}_t$ and v_t follow the discounting relationship in Equation* (5.8) *and Equation* (5.9)*. Given Σ and $(m_0, C_0, s_0, n_0, d_0)$, the forward filtering algorithm for the DLM in Equation* (5.1) *is as follows:*

With m_0, C_0, s_0, n_0, d_0,

(a). Posterior at $t-1$: $(\Phi_{t-1} \mid D_{t-1}) \sim N(m_{t-1}, C_{t-1})$.

(b). Prior at t: $(\Phi_t \mid D_{t-1}) \sim N(a_t, R_t)$, with

$$a_t = m_{t-1}, R_t = C_{t-1}/\delta_2 .$$

(c). One-step forecast: $(\boldsymbol{y}_t \mid D_{t-1}) \sim N(\boldsymbol{f}_t, Q_t)$, with

$$\boldsymbol{f}_t = F_t^{'} a_t = F_t^{'} m_{t-1}; Q_t = F_t^{'} C_{t-1} F_t/\delta_2 + s_{t-1}\Sigma .$$

(d). Posterior at t: $(\Phi_t \mid D_t) \sim T_{n_t}(m_t, C_t)$, and, $(V_t^{-1} \mid D_t) \sim G(n_t/2, d_t/2)$, with

$$A_t = R_t F_t Q_t^{-1} = C_{t-1} F_t Q_t^{-1}/\delta_2 ,$$

where $m_t = m_{t-1} + A_t e_t$, $e_t = \boldsymbol{y}_t - F_t^{'} m_{t-1}$, $C_t = \frac{s_t}{s_{t-1}}\left(\frac{C_{t-1}}{\delta_2} - A_t Q_t A_t^{'}\right)$, and

$$n_t = \delta_1 n_{t-1} + n; d_t = \delta_1 d_{t-1} + s_{t-1} e_t^t Q_t^{-1} e_t . \tag{5.10}$$

Proof. See West and Harrison (1997) for (a), (b), (c). We thus need only to derive (d).

At time t, the prior for v_t^{-1} is

$$\left(v_t^{-1} \mid \mathrm{D}_{t-1}\right) \sim \mathrm{Gamma}(\delta_1 n_{t-1}/2, \delta_1 d_{t-1}/2)\,.$$

Coupled with the likelihood

$$\left(e_t \mid \mathrm{D}_{t-1}, v_t^{-1}\right) \sim \mathrm{N}(0, Q_t)\,,$$

we have

$$\pi(v_t^{-1} \mid \mathrm{D}_t) \propto \frac{1}{\mid v_t Q_t \mid^{1/2}} \exp(-\frac{s_{t-1}}{v_t}\epsilon_t^t Q_t^{-1}\epsilon_t) \times (v_t^{-1})^{\delta_1 n_{t-1}/2-1} \exp(-\delta_1 d_{t-1} v_t^{-1}/2)\,.$$

This implies

$$(v_t^{-1} \mid \mathrm{D}_t) \sim \mathrm{Gamma}\left((n+\delta_1 n_{t-1})/2, \left(\delta_1 d_{t-1} + s_{t-1}\epsilon_t^t Q_t^{-1}\epsilon_t\right)/2\right)\,.$$

Consequently, we have

$$\begin{aligned} n_t &= \delta_1 n_{t-1} + n; \\ d_t &= \delta_1 d_{t-1} + s_{t-1} e_t^t Q_t^{-1} e_t; \\ s_t &= d_t/n_t\,. \end{aligned}$$

□

Therefore, after applying the forward filtering algorithm (assuming unknown $\{v_t\}$) in Theorem 5.1, we obtain the posterior distribution of $\left(v_T^{-1} \mid \mathrm{D}_T\right)$,

$$\left(v_T^{-1} \mid \mathrm{D}_T, \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}\right) \sim \mathrm{Gamma}\left(n_T/2, d_T/2\right)\,. \tag{5.11}$$

We make the draw $v_T^{(i)}$ according to Equation (5.11), and draw $v_t^{(i)}, t = T-1, \ldots, 1$ recursively,

$$\frac{1}{v_t^{(i)}} = \frac{\delta_1}{v_{t+1}^{(i)}} + \text{Gamma}\left((1-\delta_1)n_t/2, d_t/2\right) .$$

At the end of the MCMC algorithm, we obtain the posterior draws

$$\{v_1^{(i)}, \ldots, v_T^{(i)}; \mathbf{\Phi}_1^{(i)}, \ldots, \mathbf{\Phi}_T^{(i)}; \boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}; i = 1, \ldots N\} .$$

Inference in the following section is based on this set of random draws.

5.4 Results

p is chosen to be 20 for the example data. We run the Markov Chain for 200000 iterations and keep one sample after every 100 steps. The first 1000 iterations are treated as burn-in and are therefore discarded in the following inferences. Figure 5.2 gives the trace plot, prior distribution (up to a normalizing constant), posterior distribution, and autocorrelation function for β. For the purpose of making comparisons between the prior and the posterior distribution for β, we highlight with a red line the prior distribution in the interval $(1, 2)$, within which the posterior draws are concentrated.

Suppose $\left\{\phi_{t,j}^{(i)}\right\}$ is the i'th MCMC draw for the TVAR coefficients $\{\phi_{t,j}\}$. We calculate the posterior mean $\hat{\phi}_{t,j}$ and $\hat{v}_t$ by

$$\hat{\phi}_{t,j} = \frac{1}{N}\sum_{i=1}^{N} \phi_{t,j}^{(i)}, \qquad \hat{v}_t = \frac{1}{N}\sum_{i=1}^{N} v_t^{(i)} .$$

Figure 5.3 and Figure 5.4 show the point-wise posterior means for $\phi_{t,j}^{(i)}$ and $\{v_t\}$, respectively.

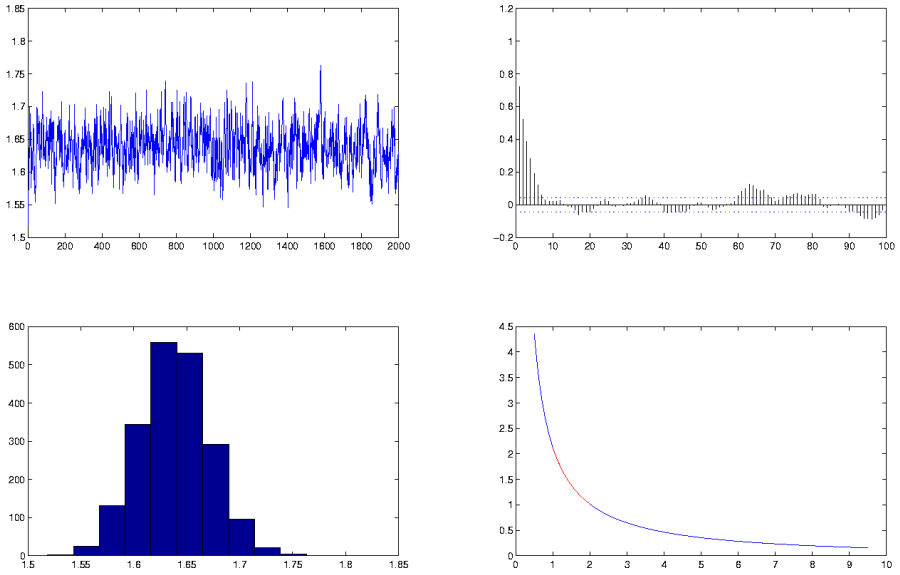

Figure 5.2: Upper-left: trace plot of the MCMC samples for β; Upper-Right: autocorrelation functions of the MCMC samples for β; Lower-Left: posterior distribution of β; Lower-Right: prior density of β.

The prediction of the computer model output at an untried input value $\boldsymbol{x}$ is obtained by spatial interpolation. Letting $e_t(\boldsymbol{z}_i) = y_t(\boldsymbol{z}_i) - \sum_{j=1}^{p} y_{t-j}(\boldsymbol{z}_i)\phi_{t,j}$ and $\boldsymbol{\rho}_{\boldsymbol{z}} = (\mathbb{C}\text{orr}(\boldsymbol{z}, \boldsymbol{z}_1), \ldots, \mathbb{C}\text{orr}(\boldsymbol{z}, \boldsymbol{z}_n))^{'}$, we have

$$(y(\boldsymbol{z}, t) \mid \boldsymbol{Y}_{t-1:t-p}(\boldsymbol{z}), \text{Data}, \{\boldsymbol{\Phi}_i, v_i, i = 1, \ldots T\}, \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}) \sim \text{N}\left(\mu_t(\boldsymbol{z}), \sigma_t^2(\boldsymbol{z})\right) . \tag{5.12}$$

The mean and the variance in the above normal distribution is given by

$$\mu_t(\boldsymbol{z}) = \sum_{j=1}^{p} y(\boldsymbol{z}, t-j)\phi_{t,j} + v_t^{-1}\boldsymbol{\rho}_{\boldsymbol{z}}^{'}\Sigma^{-1}\begin{pmatrix} e_t(\boldsymbol{z}_1) \\ \vdots \\ e_t(\boldsymbol{z}_n) \end{pmatrix} ,$$

and

$$\sigma_t^2(\boldsymbol{z}) = v_t\left(1 - \boldsymbol{\rho}_{\boldsymbol{z}}^{t}\Sigma^{-1}\boldsymbol{\rho}_{\boldsymbol{z}}\right) .$$

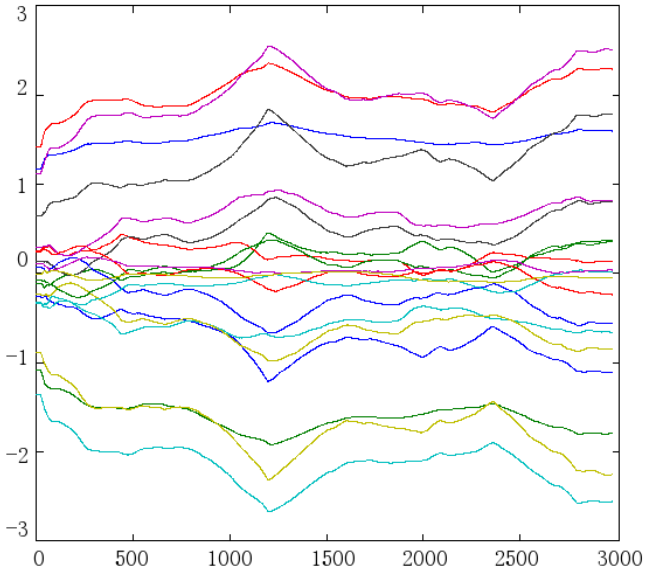

Figure 5.3: Posterior means for the `TVAR` coefficients $\{\phi_{t,j}\}$.

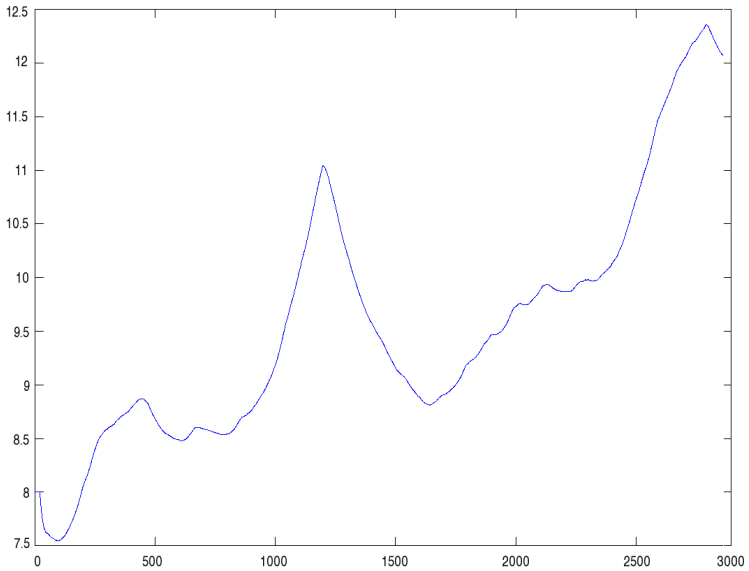

Figure 5.4: Posterior means for the time varying variances $\{v_t\}$

Equation (5.12) defines an emulator for the computer model. This emulator is an interpolator. If $\boldsymbol{z} \in \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$, we have $\mu_t(\boldsymbol{z}) = y_t(\boldsymbol{z})$ and $\sigma_t^2(\boldsymbol{z}) = 0$.

Figure 5.5 gives the result for emulating the computer model output at $\boldsymbol{z}_0 = 0.5$, with the example data shown in Figure 5.1. We only show the result at the time intervals $(1100, 1300)$ and $(2700, 2900)$, since the data is exhibiting interesting features. The posterior predictive curve is colored in green, with pointwise 90% confidence bands plotted in red. For comparison, we plot the original data as the blue curve in the picture. The prediction given by the DLM emulator indeed is quite close to the real data.

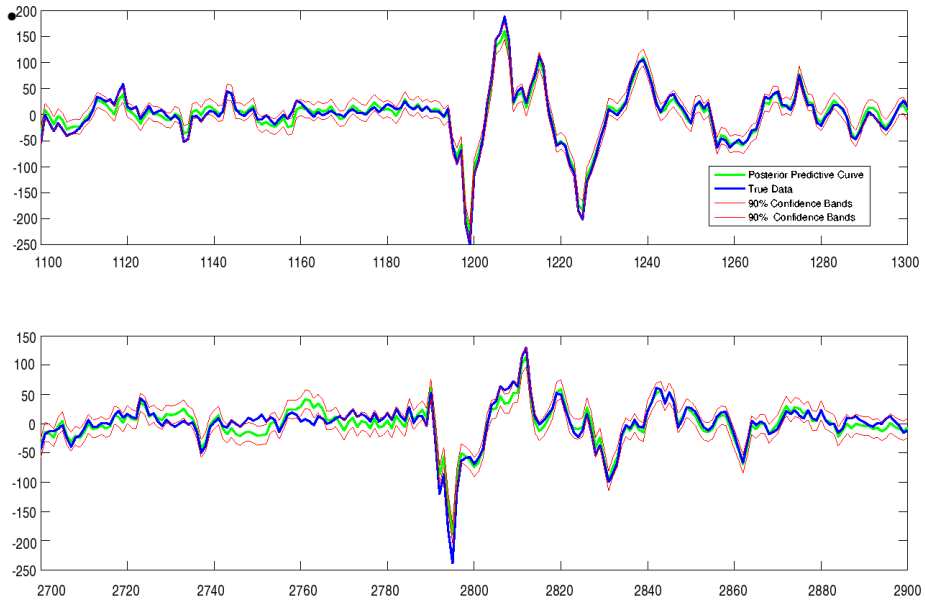

Figure 5.5: Posterior predictive curve (green), true computer model output (red), and 90% piece-wise predictive intervals for spatial interpolation with input value $\boldsymbol{x} = 0.5$.

Appendix A

Kronecker Product

The Kronecker product of two matrices $A = (a_{ij})_{i,j}$ and $B = (b_{ij})_{i,j}$ is defined as,

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

It has the following properties.

1. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$ if A and B are both invertible.

2. $|A \otimes B| = |A|^{d_2}|B|^{d_1}$, where d_1 and d_2 are the dimensions of A, B.

3. $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ if the dimensions are matched.

   If we assume $A = U_1 U_1^{'}$, $B = U_2 U_2^{'}$, then $(A \otimes B) = (U_1 \otimes U_2)(U_1 \otimes U_2)^{'}$.

Appendix B

Data Registration and Wavelet Decomposition in Chapter 3

For the wavelet representations of the output curves it will be important for the same wavelet basis elements to simultaneously represent the important features of all of the curves. In the test bed problem, the heights of the peaks and valleys of the curves from the field data are of primary importance, and their locations are not the same across the curves, due to random fluctuations in the tests. Thus we first *align* the curves so that the major peaks and the major valleys occur at the same location. In other applications, alignment would likely be based on other key features of the curves. (In the test bed, the timing of the major events is not of concern - only the forces at these events are of interest. If it were important for the computer model to accurately reflect timing, as in the analysis of airbag deployment in Bayarri *et al.* (2005b), this mode of registration could not be used.)

We did not try to align the curves from the computer model runs, since variation in these curves could not be ascribed to random fluctuation. (One might think that the computer model curves would be automatically aligned but, surprisingly, in the test bed they did show some misalignment, perhaps due to differences in the damping parameters.) We construct a reference curve (for alignment of the field curves) by averaging the model-run curves and use piecewise linear transformation to align the peaks and valleys of the field curves to this reference curve. The details are as follows:

Step 1. Construct a dyadic grid (points of the form $j/2^q$ on the interval where the function is defined. For the test bed, the interval [0, 65] covered the range of importance and $q = 12$). For each computer run extract the output values on the dyadic grid. Construct a pseudo output for points not in the grid by linear interpolation. Henceforth treat the pseudo outputs as if they are the actual outputs.

Step 2. From the K computer runs (there $K = 65$ in the test bed, define the reference curve as $\bar{y}^M(t) = \frac{1}{K}\sum_{k=1}^{K} y^M(\boldsymbol{x}_k, \boldsymbol{u}_k; t)$.

- For the first major event, located in the region $6 < t < 11$, define
  - A = location (value of t) of the maximum of the reference curve $\bar{y}^M(t)$;
  - A_r^F = location of the maximum of y_r^F;
  - a = location of minimum of $\bar{y}^M(t)$;
  - a_r^F = location of minimum of y_r^F;
- For the second major event, located in the region $37 < t < 41$, define B, B_r^F, b, b_r^F analogously. Assume $a < A < b < B$.

Step 3. For each r, match a_r^F, A_r^F with a, A by transforming t in $[a_r^F, A_r^F]$ to

$$t' = a + (t - a_r^F)\frac{A - a}{A_r^F - a_r^F}.$$

Now define the registered y_r^F on the interval $[a, A]$ as *registered* $y_r^F(t') =$ *original* $y_r^F(t)$, where t' and t are connected as above.

Step 4. Assume that $b < B$. As in Step 3, register y_r^F on the interval $[A_r^F, b_r^F]$ by mapping $[A_r^F, b_r^F]$ into $[A, b]$ via $t' = A + (t - A_r^F)\frac{b-A}{b_r^F - A_r^F}$. Similar registrations are done for the intervals $[0, a_r^F]$, $[b_r^F, B_r^F]$ and $[B_r^F, 65]$.

Figure 3.1 shows the registered data for Site 1 of the suspension system. "Time" in the figures is not literal time, but is a convenient scaling of the realigned time.

The wavelet decompositions are described in part in Section 3.2.2. The basis functions are such that at "level" 0 there is a scaling constant and for level $j = 1, \ldots, 12$ there are 2^{j-1} basis functions. To balance the need for approximation accuracy with the need to minimize the number of terms for computational feasibility, we considered each model-run and field curve and retained all coefficients at levels 0 through 3; for levels $j > 3$, we retained those coefficients that, in magnitude, were among the upper 2.5% of *all* coefficients at *all* levels for the given function, according to the R *wavethresh* thresholding procedure. We then took the union of all resulting basis functions for all the model-run and field curves. For the test bed there were 231 retained elements for the output from Site 1 on the system, and 213 for the output from Site 2. The combined (from both sites) number of retained elements was 289 and we used these for all analyses.The indices attached to these 289 retained basis elements are denoted by I.

Appendix C

Supplementary details for Chapter 4

C.1 The Example Data

The vehicle suspension system A and vehicle suspension system F correspond to the two systems in Chapter 3. We first generate the computer model runs for system B, C, D, E conditional on the runs of system A and F.

In Chapter 3, we have obtained $\omega_i^M(\boldsymbol{x}_A, \boldsymbol{\delta}_j, \boldsymbol{u}_j)$ and $\omega_i^M(\boldsymbol{x}_F, \boldsymbol{\delta}_j, \boldsymbol{u}_j)$. The estimates of the GP parameters for system A is also available as

$$\widehat{\boldsymbol{\theta}}_i^A = \{\widehat{\mu}_i^M, \widehat{\lambda}_i^M, \widehat{\boldsymbol{\alpha}}_i, \widehat{\boldsymbol{\beta}}_i\}\,.$$

We set $\phi_i = 0.5$, $\mu_i = \widehat{\mu}_i^M$, $\sigma_i^{2M} = 1/\widehat{\lambda}_i^M$ and

$$\begin{aligned}
\beta_{ik}^u = \widehat{\boldsymbol{\beta}}_{ik}\,, \qquad \alpha_{kl}^u = \widehat{\boldsymbol{\alpha}}_{ik}\,, \qquad k = 1, 2\,,\\
\beta_{i(k-2)}^\delta = \widehat{\boldsymbol{\beta}}_{ik}\,, \quad \alpha_{i(k-2)}^\delta = \widehat{\boldsymbol{\alpha}}_{ik}\,, \quad k = 3, \ldots, 9\,,\\
\beta_{ik}^x = \phi_i \beta_{ik}^\delta\,, \qquad \alpha_{ik}^x = \alpha_{ik}^\delta\,, \qquad k = 1, \ldots, 7\,.
\end{aligned}$$

Consequently, for the i^{th} wavelet coefficients, we define the correlation functions in Equation 4.11 and Equation 4.12.

Assume that

$$\omega_i^M\left(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}\right) \sim \mathrm{GP}\left(\mu_i, \sigma_i^{2M} \mathbb{C}\mathrm{orr}_i^{M_1}(\cdot,\cdot)\mathbb{C}\mathrm{orr}_i^{M_2}(\cdot,\cdot)\right)\,. \tag{C.1}$$

Letting $\Sigma_i^{M_1}$ and $\Sigma_i^{M_2}$ be the resulting correlation matrix, $\boldsymbol{\omega}_i^S$ be the vector of $\omega_i^M(\boldsymbol{x}_S, \boldsymbol{\delta}_j, \boldsymbol{u}_j)(S = A, \ldots, F)$, and

$$\boldsymbol{\omega}_i = \left((\boldsymbol{\omega}_i^A)', (\boldsymbol{\omega}_i^B)', (\boldsymbol{\omega}_i^C)', (\boldsymbol{\omega}_i^D)', (\boldsymbol{\omega}_i^E)', (\boldsymbol{\omega}_i^F)'\right)'\,,$$

we have

$$\boldsymbol{\omega}_i \sim \mathrm{N}\left(\mu_i^M \mathbf{1}, \sigma_i^{2M} \Sigma_i^{M_1} \otimes \Sigma_i^{M_2}\right) .$$

As a result, given $\boldsymbol{\omega}_i^A$ and $\boldsymbol{\omega}_i^F$,

$$\boldsymbol{\omega}_i^{B-E} = \left((\boldsymbol{\omega}_i^B)', (\boldsymbol{\omega}_i^C)', (\boldsymbol{\omega}_i^D)', (\boldsymbol{\omega}_i^E)'\right)'$$

is distributed as a multivariate normal distribution with mean vector

$$\mu \mathbf{1}_{4n\times 1} + A_{12} A_{22}^{-1} \otimes \mathbf{I}_{n\times n} \begin{pmatrix} \boldsymbol{\omega}_i^A - \mu \mathbf{1}_{n\times 1} \\ \boldsymbol{\omega}_i^F - \mu \mathbf{1}_{n\times 1} \end{pmatrix} ,$$

and covariance matrix

$$\sigma_i^{2M} (A_{11} - A_{12} A_{22}^{-1} A_{12}') \otimes \Sigma_i^{M_2} .$$

The matrices A_{11}, A_{12}, A_{22} are sub-matrices of $\Sigma_i^{M_1}$,

$$A_{11} = \begin{pmatrix} \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^B, \boldsymbol{x}^B) & \dots & \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^B, \boldsymbol{x}^E) \\ \vdots & \ddots & \vdots \\ \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^E, \boldsymbol{x}^B) & \dots & \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^E, \boldsymbol{x}^E) \end{pmatrix} ,$$

$$A_{22} = \begin{pmatrix} \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^A, \boldsymbol{x}^A) & \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^A, \boldsymbol{x}^F) \\ \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^F, \boldsymbol{x}^A) & \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^F, \boldsymbol{x}^F) \end{pmatrix} ,$$

$$A_{12} = \begin{pmatrix} \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^B, \boldsymbol{x}^A) & \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^B, \boldsymbol{x}^F) \\ \vdots & \vdots \\ \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^E, \boldsymbol{x}^A) & \mathbb{C}\mathrm{orr}_i^{M_1}(\boldsymbol{x}^E, \boldsymbol{x}^F) \end{pmatrix} .$$

We generate the wavelet coefficients $\boldsymbol{\omega}_i^{B-E}$ from the above conditional distribution.

For the field runs, we first generate $b_i(\boldsymbol{x}^S)$ for $S \in \{A, B, C, D, E, F\}$ $(i \in I)$. We first set

$$b_i(\boldsymbol{x}^A) = \frac{1}{n_{11}} \sum_{r=1}^{n_{11}} \omega_{ir}(\boldsymbol{x}^A, \boldsymbol{\delta}_{11}^*) - \frac{1}{n} \sum_{j=1}^{n} \omega_i^M(\boldsymbol{x}^A, \boldsymbol{\delta}_j, \boldsymbol{u}_j) ,$$

and

$$b_i(\boldsymbol{x}^F) = \frac{1}{n_{61}} \sum_{r=1}^{n_{61}} \omega_{ir}(\boldsymbol{x}^F, \boldsymbol{\delta}_{61}^*) - \frac{1}{n} \sum_{j=1}^{n} \omega_i^M(\boldsymbol{x}^F, \boldsymbol{\delta}_j, \boldsymbol{u}_j)\,.$$

Letting $\boldsymbol{b}_i = (b_i(\boldsymbol{x}^A), \ldots, b_i(\boldsymbol{x}^F))'$, we assume

$$\boldsymbol{b}_i \sim \mathrm{N}\left(0, \frac{\sigma_i^{2M}}{25}\Sigma_i^{M_1}\right)\,.$$

We generate $b_i(\boldsymbol{x}^B)$, $\ldots$, $b_i(\boldsymbol{x}^E)$ from its conditional distribution given $b_i(\boldsymbol{x}^A)$ and $b_i(\boldsymbol{x}^F)$.

For each individual vehicle with nominal $\boldsymbol{x}^S$ and manufacturing variation $\boldsymbol{\delta}_{ij}^*$, we generate

$$\omega_i^B(\boldsymbol{x}^S, \boldsymbol{\delta}_{ij}^*) = b_i(\boldsymbol{x}^S) + \mathrm{N}(0, \sigma_i^{2M}/25)\,,$$

and

$$\omega_i^F(\boldsymbol{x}^S, \boldsymbol{\delta}_{ij}^*) = \frac{1}{n} \sum_{l=1}^{n} \omega_i^M(\boldsymbol{x}^S, \boldsymbol{\delta}_l, \boldsymbol{u}_l) + \omega_i^B(\boldsymbol{x}^S, \boldsymbol{\delta}_{ij}^*) + \mathrm{N}(0, \widehat{\sigma}_i^{2F})\,,$$

where

$$\begin{aligned}\widehat{\sigma}_i^{2F} = &\sum_{r=1}^{n_{11}} (\omega_{ir}(\boldsymbol{x}^A, \boldsymbol{\delta}_{11}^*) - \bar{\omega}_i(\boldsymbol{x}^A, \boldsymbol{\delta}_{11}^*))^2/(n_{11} + n_{61} - 2) \\ &+ \sum_{r=1}^{n_{61}} (\omega_{ir}(\boldsymbol{x}^F, \boldsymbol{\delta}_{61}^*) - \bar{\omega}_i(\boldsymbol{x}^F, \boldsymbol{\delta}_{61}^*))^2/(n_{11} + n_{61} - 2)\,.\end{aligned}$$

At the end, we generate $\boldsymbol{\omega}_i^{B-E}$ for the computer model runs and $\omega_i^F(\boldsymbol{x}^S, \boldsymbol{\delta}_{ij}^*)$ for the field runs ($i \in I$ and $S \in \{A, \ldots, E\}$). Reconstructing these wavelet coefficients produces the functional curves in Figure 4.1.

C.2 The Iterative Maximum Likelihood Estimate Algorithm

For simplicity, we use the following notation:

$$(\mu_k, \sigma_k^2, \boldsymbol{\xi}_k) = \left(\mu_k, \sigma_k^2, \phi_k, \{\alpha_{kl}^{\delta}, \beta_{kl}^{\delta}, l = 1, \ldots, p_1\}, \{\alpha_{kl}^{u}, \beta_{kl}^{u}, l = 1, \ldots, p_2\}\right) .$$

The algorithm has two steps. In the first step, we estimate $(\sigma_k^2, \boldsymbol{\xi}_k)$ by maximizing the integrated log-likelihood $\log \mathrm{L}^I(\sigma_k^2, \boldsymbol{\xi}_k; \boldsymbol{a}_k^M)$,

$$\begin{aligned} \log \mathrm{L}^I(\sigma_k^2, \boldsymbol{\xi}_k; \boldsymbol{a}_k^M) &\propto \log \int \mathrm{L}\left(\mu_k, \boldsymbol{\xi}_k; \boldsymbol{a}_k^M\right) \mathrm{d}_{\mu_k} \\ &= -\frac{mn-1}{2}\log(\sigma_k^2) - \frac{1}{2}\log \mid \boldsymbol{\Sigma}_k \mid -\frac{1}{2}\log \mid \mathbf{1}'\boldsymbol{\Sigma}_k^{-1}\mathbf{1} \mid -\frac{1}{2\sigma_k^2} S_{\boldsymbol{\xi}_k}^2 , \end{aligned}$$

where $S_{\boldsymbol{\xi}_k}^2 = (\boldsymbol{a}_k^M)' Q \boldsymbol{a}_k^M$, $Q = (\Sigma_1 \otimes \Sigma_2)^{-1} P$, and $P = I - \mathbf{1}(\mathbf{1}' (\Sigma_1 \otimes \Sigma_2)^{-1} \mathbf{1})^{-} \mathbf{1}' (\Sigma_1 \otimes \Sigma_2)^{-1}$. Then, in the second step, we estimate μ_k given $\hat{\boldsymbol{\xi}}_k$ by

$$\hat{\mu}_k = \frac{\mathbf{1}'\Sigma_1^{-1} \otimes \Sigma_2^{-1} \boldsymbol{a}_k^M}{\left(\mathbf{1}'\Sigma_1^{-1}\mathbf{1}\right)\left(\mathbf{1}'\Sigma_2^{-1}\mathbf{1}\right)} .$$

The first step is carried out by using Fisher's scoring method (see Berger *et al.* (2001) and Paulo (2005)). We first re-parametrize as

$$\boldsymbol{\eta}(\boldsymbol{\xi}_k) = \left(\log\left(\frac{\phi_k}{1-\phi_k}\right), \log\left(\frac{\boldsymbol{\alpha}_k - 1}{2 - \boldsymbol{\alpha}_k}\right), \log\left(\boldsymbol{\beta}_k\right), \log\left(\frac{\boldsymbol{\alpha}_k^u - 1}{2 - \boldsymbol{\alpha}_k^u}\right), \log\left(\boldsymbol{\beta}_k^u\right)\right) .$$

Fisher's scoring method yields, at the l'th iteration,

$$\begin{aligned} \boldsymbol{\eta}^{(l+1)} &= \boldsymbol{\eta}^{(l)} + \lambda \left(\mathbf{I}\left(\boldsymbol{\eta}^{(l)}\right)\right)^{-1} \frac{\partial \log \mathrm{L}^I\left(\boldsymbol{\eta}; \boldsymbol{a}_k^M\right)}{\partial \boldsymbol{\eta}}\bigg|_{\boldsymbol{\eta}=\boldsymbol{\eta}^{(l)}} , \\ \sigma^{2(l+1)} &= \frac{1}{mn-1} S_{\boldsymbol{\xi}^{(l+1)}}^2 , \end{aligned}$$

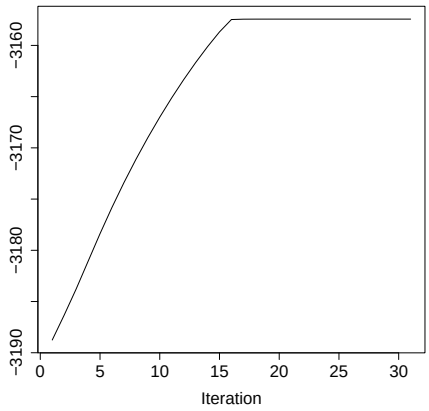

Figure C.1: The maximum log-likelihood of the first eigen-coefficient.

where $\frac{\partial \log \mathrm{L}^I\left(\boldsymbol{\eta};\boldsymbol{a}_k^M\right)}{\partial \boldsymbol{\eta}}$ and $\mathbf{I}\left(\boldsymbol{\eta}\right)$ can be computed (Berger *et al.*, 2001) as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\eta_i} \log L^I\left(\boldsymbol{\eta};\boldsymbol{y}^M\right) &= \frac{1}{2}\boldsymbol{y}^{Mt} Q W_i \boldsymbol{y}^M - \frac{1}{2}\mathrm{tr}\left(W_i\right) , \\ \boldsymbol{I}(\eta) &= \left(\frac{1}{2}\mathrm{tr}\left(W_i W_j\right)\right)_{i,j} , \end{aligned}$$

where $\Sigma = \Sigma_1 \otimes \Sigma_2$, $\dot{\Sigma}^i = \frac{\mathrm{d}}{\mathrm{d}\eta_i}\Sigma$, and $W_i = \dot{\Sigma}^i Q$. Note that $\dot{\Sigma}^i = \dot{\Sigma}_1^i \otimes \Sigma_2 + \Sigma_1 \otimes \dot{\Sigma}_2^i$. As a result, we have

$$W_i = \left(\left(\dot{\Sigma}_1^i \Sigma_1^{-1}\right) \otimes I_n + I_m \otimes \left(\dot{\Sigma}_2^i \Sigma_2^{-1}\right)\right) P .$$

We ran the algorithm for 30 iterations for every coefficient. After 30 iterations, the maximum log-likelihoods for all the coefficients have converged. Figure C.1 shows the maximum log-likelihood of the first coefficient, which stabilizes after 15 steps.

Appendix D

Supplementary details for Chapter 5

In Figure 5.1, the data with $\boldsymbol{z} = 0.5$ (in red) is obtained from a real engineering system. This data is observed at $T = 3000$ time points. We use $\boldsymbol{Y}(0.5) = (y_1(0.5), \ldots, y_T(0.5))$ to represent the whole sequence. We choose $p = 20$, $\alpha = 2$, and $\beta = 1.6$. Given $\boldsymbol{Y}(0.5)$ and its fit $\{\phi_{t,j}, v_t, j = 1, \ldots, p\}$, we simulate the data for $\boldsymbol{Y}(\boldsymbol{z})$ at $\boldsymbol{z} = 0.25, \ldots, 0.75$ as follows:

$$\begin{pmatrix} y_t(\boldsymbol{z}_1) \\ y_t(\boldsymbol{z}_2) \\ \cdots \\ y_t(\boldsymbol{z}_n) \end{pmatrix} \mid \{y_{t-1}(\boldsymbol{z}_i)\}, \{\boldsymbol{\Phi}_t\}, y_t(\boldsymbol{z}) \quad \sim \quad \text{MVN}\left(\boldsymbol{\mu}_t = \begin{pmatrix} \mu_t(\boldsymbol{z}_1) \\ \mu_t(\boldsymbol{z}_2) \\ \cdots \\ \mu_t(\boldsymbol{z}_n) \end{pmatrix}, v_t \tilde{\Sigma} \right) ,$$

where

$$\mu_t(\boldsymbol{z}_i) \quad = \quad \sum_j \phi_{t,j} Y_{t-j}(\boldsymbol{z}_i) + v_t^{-1} \boldsymbol{\rho}' \left(y(0.5, t) - \sum_j \phi_{t,j} y_{t-j}(0.5) \right) ,$$

and

$$\tilde{\Sigma} = \Sigma - \boldsymbol{\rho}\boldsymbol{\rho}' .$$

Σ is the $n \times n$ matrix with (i, j) element $(\Sigma)_{i,j} = \mathbb{C}\text{orr}(\boldsymbol{z}_i, \boldsymbol{z}_j)$. $\boldsymbol{\rho}$ is an $n \times 1$ vector with i'th element equal to $\mathbb{C}\text{orr}(0.5, \boldsymbol{z}_i)$.

Bibliography

Aslett, R., Buck, R. J., Duvall, S. G., Sacks, J., and Welch, W. J. (1998). Circuit optimization via sequential computer experiments: Design of an output buffer. *Applied Statistics* **47**, 1, 31–48.

Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P. (1996). Experimental design and observation for large systems (Disc: p95-111). *Journal of the Royal Statistical Society, Series B, Methodological* **58**, 77–94.

Bayarri, M., Berger, J., Garcia-Donato, G., Liu, F., Palomo, J., Paulo, R., Sacks, J., Walsh, D., Cafeo, J., and Parthasarathy, R. (2006). Computer model validation with functional outputs. *Niss tech. report* .

Bayarri, M. J., Berger, J. O., Higdon, D., Kennedy, M. C., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2002). A framework for validation of computer models. Tech. Rep. 128, National Institute of Statistical Sciences, http://www.niss.org/technicalreports/tr128.pdf.

Bayarri, M. J., Berger, J. O., Kennedy, M., Kottas, A., Paulo, R., Sacks, J., Cafeo, J. A., Lin, C. H., and Tu, J. (2005b). Validation of a computer model for vehicle collision. Tech. rep., National Institute of Statistical Sciences.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2005a). A framework for validation of computer models. Tech. rep., National Institute of Statistical Sciences.

Berger, J. O., De Oliveira, V., and Sansó, B. (2001). Objective Bayesian analysis

of spatially correlated data. *Journal of the American Statistical Association* **96**, 1361–1374.

Craig, P. S., 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**, 454, 717–729.

Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1997). Pressure matching for hydrocarbon reservoirs: a case study in the use of bayes linear strategies for large computer experiments. In *Case Studies in Bayesian Statistics: Volume III. Gatsonis, C., Hodges, J. S., Kass, R. E., McCulloch, R., Rossi, P. and Singpurwalla, N. D.* 36–;93.

Currin, C., Mitchell, T., Morris, M., 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.

Dowding, K. J., Pilch, M., and Hills, R. G. (2006). Model validation challenge problems:thermal problem .

Goldstein, M. (1998). Bayesian linear analysis. In *Encyclopaedia of Statistical Sciences*, vol. 3. Wily.

Goldstein, M. and Rougier, J. C. (2003). Calibrated bayesian forecasting using large computer simulators. Tech. rep., Statistics and Probability Group, University of Durham, http://www.maths.dur.ac.uk/stats/physpred/papers/CalibratedBayesian.ps.

Goldstein, M. and Rougier, J. C. (2004). Probabilistic formulations for transferring inferences from mathematical models to physical systems. Tech. rep., Statistics and Probability Group, University of Durham, http://www.maths.dur.ac.uk/stats/physpred/papers/directSim.pdf.

Goldstein, M. and Rougier, J. C. (2006). Bayes linear calibrated prediction for complex systems. *Journal of the American Statistical Association* **101**, 1132–1143.

Higdon, D., Gattiker, J., Williams, B., and M., R. (2007). Computer model validation using high dimensional outputs. In J. Bernardo, M. J. Bayarri, A. P. Dawid, J. O. Berger, D. Heckerman, A. F. M. Smith, and M. West, eds., *Bayesian Statistics 8*. Oxford University Press, London. (in press).

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.

McKay, M. D., Conover, W. J., and Beckman, R. 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.

Morris, J., Vannucci, M., Brown, P., and Carroll, R. (2003). Wavelet-based nonparametric modeling of hierarchical functions in colon carcinogenesis (with discussion). *Journal of the American Statistical Association* **98**, 573–597.

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.

Müller, P. and Vidakovic, B. (1999). *Bayesian Inference in Wavelet Based Models*, vol. Lecture Notes in Statistics 141. Springer-Verlag: New York.

Nagy, B. (2006). Fast bayesian implementation (fbi) of gaussian process regression. Presentation at Workshop on Complex Computer Models, Simon Fraser University.

Oakley, J. (2004). Estimating percentiles of computer code outputs. *Applied Statistics* **53**, 83–93.

Oakley, J. and O'Hagan, A. (2002). Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika* **89**, 4, 769–784.

Oakley, J. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of the Royal Statistical Society B* **66**, 751–769.

Paulo, R. (2005). Default priors for Gaussian processes. *Annals of Statistics* **33**, 2, 556–582.

Ramsay, J. and Silverman, B. (1997). *Functional Data Analysis*. Springer.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments (C/R: p423-435). *Statistical Science* **4**, 409–423.

Saltelli, A., Chan, K., and Scott, M., eds. (2000). *Sensitivity Analysis*. Wiley, Chichester.

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

Schonlau, M. and Welch, W. (2005). Screening the input variables to a computer code via analysis of variance and visualization. In A. Dean and S. Lewis, eds., *Screening*. Springer-Verlag, New York.

Vidakovic, B. (1999). *Statistical Modeling by Wavelets*. Wiley: New York.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics* **34**, 15–25.

West, M. and Harrison, P. (1997). *Bayesian Forecasting and Dynamic Models*. Springer, New York, USA.

Biography

I was born in Heilongjiang, China, on the 9th of September 1979. I graduated from Beijing University (aka Peking University) with a bachelor degree in the year 2002. In 2002, I enrolled in the Ph.D. program at ISDS and became a Ph.D. candidate in the spring of 2004.