PREDICTION AND CALIBRATION USING OUTPUTS FROM MULTIPLE COMPUTER SIMULATORS

by

Joslin Goh

M.Sc., Simon Fraser University, 2009
B.Math., University of Waterloo, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy
in the
Department of Statistics and Actuarial Science
Faculty of Applied Sciences

© Joslin Goh 2014

SIMON FRASER UNIVERSITY
Summer 2014

All rights reserved. However, in accordance with the Copyright Act of Canada, this work may be reproduced, without authorization, under the conditions for Fair Dealing. Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.
APPROVAL

Name: Joslin Goh
Degree: Doctor of Philosophy
Title of Thesis: Prediction and Calibration Using Outputs from Multiple Computer Simulators

Examinining Committee: Dr. Tom Loughin
Chair

Dr. Derek Bingham, Professor
Senior Supervisor

Dr. David Campbell, Associate Professor
Supervisor

Dr. Richard Lockhart, Professor
Supervisor

Dr. Tim Swartz, Professor
Internal Examiner

Dr. David Higdon, Scientist
External Examiner, Los Alamos National Lab, NM

Date Approved: August 18th, 2014
Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the non-exclusive, royalty-free right to include a digital copy of this thesis, project or extended essay[s] and associated supplemental files ("Work") (title[s] below) in Summit, the Institutional Research Repository at SFU. SFU may also make copies of the Work for purposes of a scholarly or research nature; for users of the SFU Library; or in response to a request from another library, or educational institution, on SFU’s own behalf or for one of its users. Distribution may be in any form.

The author has further agreed that SFU may keep more than one copy of the Work for purposes of back-up and security; and that SFU may, without changing the content, translate, if technically possible, the Work to any medium or format for the purpose of preserving the Work and facilitating the exercise of SFU's rights under this licence.

It is understood that copying, publication, or public performance of the Work for commercial purposes shall not be allowed without the author’s written permission.

While granting the above uses to SFU, the author retains copyright ownership and moral rights in the Work, and may deal with the copyright in the Work in any way consistent with the terms of this licence, including the right to change the Work for subsequent purposes, including editing and publishing the Work in whole or in part, and licensing the content to other parties as the author may desire.

The author represents and warrants that he/she has the right to grant the rights contained in this licence and that the Work does not, to the best of the author’s knowledge, infringe upon anyone's copyright. The author has obtained written copyright permission, where required, for the use of any third-party copyrighted material contained in the Work. The author represents and warrants that the Work is his/her own original work and that he/she has not previously assigned or relinquished the rights conferred in this licence.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2013
Abstract

Computer simulators are widely used to describe and explore physical processes. In some cases, several simulators, which can be of different or similar fidelities, are available for this task. A big part of this thesis focuses on combining observations and model runs from multiple computer simulators to build a predictive model for the real process. The resulting models can be used to perform sensitivity analysis for the system, solve inverse problems and make predictions. The approaches are Bayesian and are illustrated through a few simple examples, as well as a real application in predictive science at the Center for Radiative Shock Hydrodynamics at the University of Michigan. Although the computer model can be viewed as an inexpensive way to gain insight into the physical process, it can become computationally expensive continuously exercise the computer simulator. A sequential design strategy is proposed to minimize the total number of function evaluations for finding the global extremum. The practical implementation of the proposed approach is addressed and applied to several examples containing multiple computer codes.

Keywords: Bayesian; Computer Experiments; Hierarchical Model; Markov Chain Monte Carlo; Sequential Design.
Dedication

To Chin Chew York
Acknowledgments

I would like to thank the Center of Radiative Shock and Hydrodynamics at the University of Michigan for sharing their data with me. It has been an eye-opening experience for me to learn first-hand on working in an interdisciplinary team.

I cannot express my gratitude enough for my senior supervisor, Dr. Derek Bingham, for his guidance, support and encouragement. I will always be grateful for the research opportunities that he has given me throughout the years. Many thanks to my committee member, Dr. David Higdon, Dr. Tim Swartz, Dr. Richard Lockhart and Dr. David Campbell for their insightful comments and suggestions that led to significant improvement of this dissertation.

The last seven years in the department and in this city has been amazing. There are many other people who certainly deserve thanks for helping me in every possible way. So, to you who is reading this, thank you.

To my mum and dad, thank you for your love and continuous support. I made it!
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval</td>
<td>ii</td>
</tr>
<tr>
<td>Partial Copyright License</td>
<td>iii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iv</td>
</tr>
<tr>
<td>Dedication</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>vi</td>
</tr>
<tr>
<td>Contents</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>x</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 Background</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Bayesian Hierarchical Model in Computer Experiments</td>
<td>6</td>
</tr>
<tr>
<td>2.1.1 Prior Model Formulation</td>
<td>7</td>
</tr>
<tr>
<td>2.1.2 Prior distributions and MCMC</td>
<td>9</td>
</tr>
<tr>
<td>2.1.3 Prediction at An Unknown Input Setting</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Problems Addressed</td>
<td>12</td>
</tr>
</tbody>
</table>
# 3 Multi-fidelity Model Calibration

3.1 The Hierarchical Model ........................................ 14

3.1.1 Gaussian Process Models .................................. 16

3.1.2 Prior distributions and MCMC ............................... 18

3.2 Prediction ......................................................... 20

3.3 Examples ......................................................... 20

3.3.1 Toy Example .................................................. 21

3.3.2 CRASH ......................................................... 27

3.4 Discussion ....................................................... 31

# 4 An Alternative Model for Model Calibration

4.1 Model Formulation .............................................. 36

4.1.1 Gaussian Processes ......................................... 37

4.1.2 Prior distributions and MCMC ............................... 38

4.2 Examples ......................................................... 40

4.2.1 A perfect computer model ................................... 41

4.2.2 A more challenging example ................................. 44

4.2.3 CRASH ......................................................... 48

# 5 Computer Codes With No Ranking

5.1 The Bayesian Hierarchical Model .............................. 53

5.1.1 Gaussian Processes ......................................... 55

5.2 Examples ......................................................... 56

5.2.1 A Realistic Example ......................................... 56

5.2.2 Computer Simulators When One Model is Better Than Another ........................................ 61

5.3 Discussion ....................................................... 64

# 6 Sequential Kriging Optimization

6.1 The Improvement Function ..................................... 66

6.2 The Implementation ............................................. 70

6.3 Examples ......................................................... 72
6.3.1 A simple example .............................................. 72
6.3.2 Global minima in different locations ......................... 75
6.3.3 Global minimum in an unsampled region ....................... 79
6.3.4 A bad low fidelity model ...................................... 81
6.3.5 Computer simulators of similar fidelities .................... 83
6.4 Discussion ..................................................... 86

7 Conclusion ................................................... 88

Bibliography ................................................... 90
List of Tables

6.1 A summary of the new inputs added to the systems at each iteration. The high fidelity simulator is denoted with ‘H’ while the low fidelity simulator is represented with ‘L’. ................................................................. 75
6.2 A summary of the new inputs added to the systems at each stage. The high fidelity simulator is denoted with ‘H’ while the low fidelity simulator is represented with ‘L’. The horizontal line denotes the cut-off determined by the default stopping criterion. ................................................................. 78
6.3 The new inputs added to the systems at each iteration. The high fidelity simulator is denoted with ‘H’ while the low fidelity simulator is represented with ‘L’. The horizontal line denotes the cut-off determined by the default stopping criterion. ................................................................. 81
6.4 Proposed new runs added to the simulators. ..................................................... 83
6.5 Proposed new runs added to the simulators. ..................................................... 86
List of Figures

3.1 Response surface of the low fidelity simulator, the high fidelity simulator and
the mean of the physical process as outlined in (3.10)-(3.12). .......................... 22
3.2 Predicted versus actual field measurements of the validation set (with the
$y = x$ line) .................................................................................................................. 23
3.3 Diagnostics plots for the simple example: (a) Prediction error against predictions;
(b) Prediction error against $x_1$; (c) Prediction error against $x_2$. ........... 24
3.4 Plot of the two-dimensional marginals for the posterior distributions of the
three parameters for different sample sizes. The diagonals show the marginal
posterior distributions of the calibration parameters, with the true values
marked with vertical dashed lines. The off-diagonals sub-plots contain the
two-dimensional marginal posterior distributions for the three calibration pa-
rameters. The solid lines represent the 95% high posterior density region. ... 25
3.5 Boxplots of the RMSPE obtained from the 100 simulated datasets analyzed
using models ML, MH and MLH. .......................................................... 26
3.6 A pictorial version of the apparatus used in the radiative shock experiments.
The vertical bar represents the beryllium disk where the laser deposits energy.
The shock wave breaks through the beryllium disk and moves down the xenon
filled tube (horizontal bar). .......................................................... 28
3.7 Diagnostic plots for the breakout time of the radiative shock experiment:
(a) Prediction error against predictions; (b) Prediction error against $x_1$; (c)
Prediction error against $x_2$. .......................................................... 30
3.8 Predicted versus actual breakout times and 95% prediction intervals. ... 31
3.9 The horizontal dotted line in each subplot draws the actual observed breakout time. The first two bars of ‘+’ on the far left of the plots are the simulated outputs from 1D and 2D-CRASH, respectively. The 95% prediction intervals, denoted as vertical intervals, are obtained from fitting models ML, MH and MLH, respectively. The mean of each prediction interval is denoted with circle.

3.10 Plot of the two-dimensional marginals for the posterior distribution of the four parameters. The diagonals show the marginal posterior distributions of the calibration parameters. The off-diagonals sub-plots contain the two-dimensional marginal posterior distributions for the four calibration parameters. The solid lines represent the 95% high posterior density region.

4.1 Predicted versus actual observed outputs of the validation set. The straight line is $y = x$.

4.2 Diagnostic plots: (a) Prediction error against predictions; (b) Prediction error against $x_1$; (c) Prediction error against $x_2$.

4.3 Marginal posterior distributions of the calibration parameters using the new model and commonly used model proposed by Kennedy and O’Hagan (2001).

4.4 Boxplots of the RMSPE obtained from the 100 simulated datasets analyzed using the new model and the model proposed by Kennedy and O’Hagan (2001). Each dataset contains 30 outputs from the unbiased computer simulator and 5 observations.

4.5 Predicted versus actual observed outputs of the validation set. The straight line is $y = x$.

4.6 Diagnostic plots for this slightly realistic toy example: (a) Prediction error against predictions; (b) Prediction error against $x_1$; (c) Prediction error against $x_2$.

4.7 Marginal posterior distributions of the calibration parameters using the new model and the commonly used model proposed by Kennedy and O’Hagan (2001).
4.8 Boxplots of the RMSPE obtained from the 100 simulated datasets analyzed using the new model and the model proposed by Kennedy and O’Hagan (2001). Each dataset contains 30 outputs from the biased computer simulator and 5 observations.

4.9 Predicted versus actual observed outputs of the validation set. The straight line is \( y = x \).

4.10 The horizontal dotted line in each subplot draws the actual observed breakout time. The first bar of ‘+’ on the left of the plots are the simulated outputs from 2D-CRASH. The 95% prediction intervals, denoted as vertical intervals, are obtained from fitting models MH and the proposed model, respectively. The mean of each prediction interval is denoted with circle.

4.11 Plot of the two-dimensional marginals for the posterior distribution of the three parameters. The diagonals show the marginal posterior distributions of the calibration parameters. The off-diagonals sub-plots contain the two-dimensional marginal posterior distributions for the three calibration parameters. The solid lines represent the 95% high posterior density region.

5.1 Surfaces of the Goldstein and Price function and the two computer codes.

5.2 Error of prediction versus prediction and input settings.

5.3 RMSPE of the simulation study.

5.4 Average length of the 95% credible intervals obtained from the simulation study.

5.5 Posterior distributions of \( \theta \) obtained from different models. The first row is the results of models with 5 observations while the second row shows the posterior distributions of models with 10 field measurements. The vertical line is \( \theta = 0.4 \), the actual value of the calibration parameter.

5.6 Error of prediction versus prediction and input settings.

5.7 RMSPE of the simulation study.

5.8 Average length of the 95% credible intervals obtained from the simulation study.
5.9 Posterior distributions of $\theta$ obtained from different models. The first row is the results of models with 5 observations while the second row shows the posterior distributions of models with 10 field measurements. The vertical line is the actual value of the calibration parameter.

6.1 The blue line is the function of the field experiment. The surfaces of the high and low fidelity simulators are shown in red and green lines. The 3 observations are in blue circles, whereas the red and green $\times$’s are the design variables of the initial designs for the high and low fidelity simulators. The red and green circles indicate the new $x$ added to the high and low fidelity simulated system, respectively.

6.2 Response surfaces of the field process (blue), high fidelity (red) and low fidelity (green) model. The observations are in blue circles. The red and green $\times$’s are the design variables of the initial designs for the high and low fidelity simulators. The red and green circles at the bottom of the figure indicate the new $x$ added to the high and low fidelity simulated system, respectively, under the default stopping criterion.

6.3 Response surfaces of the field process (blue), high fidelity (red) and low fidelity (green) model. The black $\Box$’s are predictions obtained under model M4 while the magenta $+$’s are predicted observations using model M20.

6.4 Response surfaces of the field process (blue), high fidelity (red) and low fidelity (green) model. The observations are in blue circles, whereas the red and green $\times$’s denote the design variables of the initial designs for the high and low fidelity simulators. The red and green circles at the bottom of the figure indicate the new $x$ added to the high and low fidelity simulated system, respectively, under the default stopping criterion.
6.5 The blue solid line is the function of the field experiment. The surfaces of the high and low fidelity simulators are shown in red and green dotted lines. The observations are in blue circles, whereas the red and green ×’s are the design variables of the initial designs for the high and low fidelity simulators. The red and green circles at the bottom of the figure indicate the new $x$ added to the high and low fidelity simulated system, respectively.

6.6 The blue solid line is the function of the field experiment. The surfaces of the first and second computer codes are in green and red. The observations are in blue circles, whereas the red and green ×’s are the design variables of the initial designs for the high and low fidelity simulators. The green and red circles at the bottom of the figure indicate the new $x$ added to the first and fidelity simulated system, respectively.
Chapter 1

Introduction

Deterministic computer models are used to simulate a wide variety of physical processes (Sacks et al., 1989; Santner et al., 2003). Oftentimes, a single run of the code requires considerable computational effort, making it infeasible to continually exercise the simulator. Instead, experimenters attempt to explore the computer model response (and to some extent the physical process) using a limited number of computer model runs.

In some applications, several simulators of the physical process are available, each with different levels of fidelity. The varying levels of fidelity can occur, for example, because of the presence of reduced order physics in lower fidelity models, different levels of accuracy specified for numerical solvers or solutions obtained on finer grids. In these cases, a higher fidelity model is assumed to better represent the physical process than a lower fidelity model, but also takes more computer time to produce an output than a lower fidelity model. Combining relatively cheap lower fidelity model runs with more costly high fidelity runs to emulate the high fidelity model has, thus, been a problem of significant interest (Kennedy and O’Hagan, 2000; Qian et al., 2006; Qian and Wu, 2008; Cumming and Goldstein, 2009).

The most common approach to combining the outputs of multi-fidelity simulators was proposed by Kennedy and O’Hagan (2000). Their work writes a high fidelity model as a
linear combination of the next lowest fidelity model and a discrepancy term. With an alter-
ation of the linear combination used to model the high fidelity simulator, Qian et al. (2006)
and Qian and Wu (2008) also used the Bayesian hierarchical Gaussian process to model the
response surfaces. Cumming and Goldstein (2009) further abstracted the model and used
a Bayes linear approach.

Another important application of computer simulators is that of calibration (e.g., Kennedy
and O’Hagan (2001); Higdon et al. (2004)) where the aim is to combine simulator outputs
with physical observations to build a predictive model and also estimate unknown parameters
that govern the behaviour of the mathematical model. The latter endeavour amounts to
solving a sort of inverse problem, while the former activity is a type of regression problem.
In this setting, it is common to write the physical observations as a sum of the simulator
output, a systematic discrepancy and observational error. This approach has been adapted
to consider a variety of output data-structures (e.g., Higdon et al. 2008; Paulo et al. 2012).
These approaches to model calibration have used only a single computer model and have
not considered the use of multi-fidelity simulators.

The first part of this work is motivated by applications at the Center for Radiative
Shock Hydrodynamics (CRASH) at the University of Michigan. The aim is to develop new
methodology to combine outputs from simulators with different levels of fidelity and field
observations to make predictions of the physical system with associated measurements of
uncertainty. The CRASH simulators also require estimation of optimal values for several
input parameters (i.e. calibration parameters), and the simulators have different calibration
parameters depending on the level of fidelity, thereby complicating the calibration problem.
In a spirit similar to Kennedy and O’Hagan (2000, 2001) and Higdon et al. (2004), we
propose a predictive model that incorporates computer model outputs and field data, while
attempting to estimate the calibration parameters. The approach calibrates each computer
model to the next highest level of fidelity model, and the simulator of the highest fidelity is
then calibrated to the field measurements. All response surfaces are modelled using Gaussian process (GP) models, and the various sources of information that inform predictions of the physical system are combined with a Bayesian hierarchical model.

This methodology is only used where there is a ranking of the computer models (e.g., low to high fidelity). However, in many settings there is no obvious ranking for the models. The lack of ranking can be due to various reasons such as non-existence of suitable ranking criteria and different understanding of the physical process. This situation is common in climate modelling and experts have combined outputs from the different simulators to make predictions (Giorgi and Mearns, 2002; Tebaldi and Knutti, 2007; Smith et al., 2009). The effort to combine the outputs from multiple simulators is known as multi-model ensembles. There are many ways to put the outputs from different sources together. The reliable ensemble average proposed by Giorgi and Mearns (2002) rewards models with small bias and projections that agree with the “consensus” of the ensembles, while penalizing models that perform poorly or contain outliers. Tebaldi et al. (2004, 2005) and Smith et al. (2009) introduced Bayesian models that treat the quantities of interest (e.g. current and future climate) as random variables and regard the statistical distribution of the data as a function of the unknown parameters that govern the likelihood function. In general, it is felt that combining simulations from multiple models can increase reliability and consistency of model forecasts. It has been reported to be superior to a single model prediction in different applications such as public health (Thomson et al., 2006), agriculture (Cantelaube and Terres, 2005) and climate modeling (Tebaldi and Knutti, 2007).

On the other hand, some of the multi-model ensembles are known to produce unrealistically low estimates of uncertainty in future model projection (Tebaldi et al., 2005) whereas some rely too heavily on the comparison between simulated and observed responses (Greene et al., 2006; Tebaldi et al., 2004). By modifying the model proposed by Kennedy and O’Hagan (2001), a new methodology is introduced in the second part of this dissertation to overcome the issues encountered in multi-model ensembles. The new approach
incorporates information from the field measurements and simulation runs in a Bayesian hierarchical model and estimates the calibration parameters by calibrating all the computer models to the actual system simultaneously. The response surfaces are also modelled through independent GPs and predictions can be carried out using realizations from the posterior distributions of the calibration and statistical parameters.

Calibration and prediction are not the only objectives in computer experiments. Experts are, sometimes, interested to identify features (e.g. global minimum and maximum) of the response surfaces. Schonlau and Welch (1996) proposed the expected improvement (EI) algorithm to search for the global extremum through a sequential design strategy. The EI algorithm aims at minimizing the total number of function evaluations for finding the global extremum of a deterministic function and has been widely used in computer experiment applications (e.g. Jones et al. (1998)). Huang et al. (2006a) and Huang et al. (2006b) extended the algorithm to explore multi-fidelity response surfaces while Forrester et al. (2007) modified the efficient global optimization method to confirm and enhance the predictions of the computer models.

The deterministic functions discussed in the literature of global optimization usually do not require the estimation of calibration parameters. However, the computer models of interest in this work contain calibration parameters, making the existing global optimization not suitable. The last part of the thesis focuses on extending the EI algorithm (Schonlau and Welch, 1996; Jones et al., 1998) and augmented EI algorithm (Huang et al., 2006a) for multi-fidelity computer simulators containing calibration parameters. The goal, however, is not the extremum itself but its location. The proposed algorithm is a sequential design strategy based on the optimization of a probabilistically based improvement function. In practice, reasons involving costs, nature of the process or experimental venue result in limitations on the number of simulation trials. An additional feature is built into the algorithm to identify the most cost efficient computer simulator, while maximizing the information gain from the new simulation run at each stage.
Before delving into the results of this thesis, the model proposed by Kennedy and O’Hagan (2001) will be presented in Chapter 2. GP regression will also be outlined, providing sufficient background for the chapters that follow.

The proposed methodology in Chapter 3 extends the model proposed by Kennedy and O’Hagan (2001) and Higdon et al. (2004) for multi-fidelity computer models. The relevant prior distributions of the GPs used to model the response surfaces will be presented together with the framework for making predictions. A simple example from the literature and an application from CRASH are used to demonstrate the proposed approach. Further discussion such as limitations of this model can be found at the end of the chapter.

In Chapter 4, we take the view that there exists a perfect simulator that can model the field process flawlessly. Such belief allows the construction of a new predictive model that attempts to calibrate the parameter of interest with a limited number of observations. A few simple examples are presented to illustrate the new model. Further evaluation of the prediction and calibration performance is done using an application from CRASH.

We switch the focus back to multiple computer simulators in Chapter 5. The proposed approach extends the model in Chapter 4 for applications with numerous simulated systems that cannot be ranked. The GPs used to model the response surfaces are briefly mentioned here as they are analogous to the framework in the previous chapters. An example from the literature is modified in two scenarios to assess the prediction and calibration performance of the new model before discussing the drawbacks of the new model and future research directions.

The extension of the EI algorithm (Schonlau and Welch, 1996) is discussed in Chapter 6. The proposed modification of the EI algorithm is presented along with the additional criteria used to select the most cost efficient computer model at each stage. A few simple
examples with multiple computer simulators are discussed to demonstrate the execution of
the new strategy in different scenarios. Chapter 7 summarizes the new frameworks presented
in this work and concludes the dissertation with potential future research in applications
with multiple computer simulators.
Chapter 2

Background

In this chapter, we will review the framework for computer model calibration that was proposed by Kennedy and O’Hagan (2001) to fine tune computationally demanding computer simulators that may be biased when compared to reality. They attempt to correct the inadequacy of the computer model through a discrepancy between simulation runs and observations. Moreover, their predictive model allows for all sources of uncertainty, including the uncertainty for the model parameters. Their implementation was later modified by Higdon et al. (2004) into a single Bayesian hierarchical model and has been widely used in many applications (e.g. Bayarri et al. (2007); Reese et al. (2004); Linkletter et al. (2006); Williams et al. (2006)).

2.1 Bayesian Hierarchical Model in Computer Experiments

Throughout, the computer models are assumed to be deterministic mathematical functions that rely on two types of inputs: design variables, denoted as $x = (x_1, \ldots, x_p)$, and calibration parameters, denoted as $t = (t_1, \ldots, t_m)$. The latter inputs can only be adjusted within the simulator, but are not measurable in the field. We use $t$ to denote inputs for calibration parameters used to run the code and let $\theta$ stand for the true, or optimal, values of the calibration parameters in the field. The goal of the model calibration problem is to build a predictive model for the field process while estimating the unknown calibration parameters.
A univariate observation of the physical process, $y^f(\cdot)$, is commonly modelled as a biased and noisy version of the computer model, $\eta^c(\cdot)$. At input setting $(x, t)$, denote the simulated output as

$$y^c(x, t) = \eta^c(x, t),$$  \hfill (2.1)

and a field measurement as

$$y^f(x) = \eta^c(x, \theta) + \delta(x) + \epsilon.$$  \hfill (2.2)

The term $\delta(\cdot)$ accounts for the systematic discrepancy between the simulated and physical systems. The form of $\delta(\cdot)$ can be varied based on experts’ knowledge. The experimenters can usually characterize the size and nature of the measurement errors, $\epsilon$, and provide insights to their distributions. In this work, the measurement errors are independent and follow a Normal distribution with mean 0 and variance $\frac{1}{\lambda_y}$. The parameter $\lambda_y$ is commonly known as the marginal precision parameter.

### 2.1.1 Prior Model Formulation

To make predictions of the physical system, the response surfaces for the simulator and discrepancy need to be estimated. We follow the common practice of using independent GPs to model the response surfaces (e.g., see Sacks et al., 1989; Kennedy and O’Hagan, 2001). The reason for using these models, in general, boils down to the success of the GP as a non-parametric regression estimator and also the ability of the GP model to provide a basis for statistical inference for the outputs of deterministic computer codes. From a Bayesian viewpoint in this context, one can think of the GP as a prior distribution over the class of functions coming from the computer simulator and the discrepancy, respectively.

Specifically, the simulated outputs are treated as a realization of a random function of the form:

$$Y^c(x, t) = \sum_{i=1}^{p} f_i(x, t) \beta_i + Z(x, t),$$  \hfill (2.3)
where \( f_1, \ldots, f_p \) are regression functions, \( \boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)' \) is the vector of unknown regression coefficients, and \( Z \) is a mean zero GP. As in most simulator applications (e.g. Higdon et al. (2004); Linkletter et al. (2006); Higdon et al. (2008)), the mean of the GP process is treated as a constant mean, \( \mu \) and the covariance function is used to model the response surface. At inputs \((x, t)\) and \((x', t')\), the covariance between the observations is given by

\[
\text{Cov} \left[ Z(x, t), Z(x', t') \right] = \frac{1}{\lambda_z} \prod_{s=1}^{p} \rho_{z,s}^{4(x_s-x'_s)^2} \prod_{s=1}^{m} \rho_{z,p+s}^{4(t_s-t'_s)^2}. \tag{2.4}
\]

The parameter \( \lambda_z \) denotes the marginal precision of the computer simulator and \( \rho_{z,s} \in [0, 1] \) denotes the dependence of the process in the direction of the \( s \)-th component. The closer \( \rho_{z,s} \) is to zero, the more complex the relationship between the response and the \( s \)-th component.

To account for numerical fluctuations commonly found in complicated computer codes, Linkletter et al. (2006) proposed to include a white noise component to the covariance function. It is noted that the covariance function in (2.4) is the same as that proposed by Sacks et al. (1989) under the reparameterization of

\[
\rho_{z,s} = \exp \left\{ -\frac{\gamma_{z,s}}{4} \right\}.
\]

The parameterization in (2.4) is preferred because it eases the exploration of the posterior of \( \lambda_z \) and \( \rho \)'s through Monte Carlo Markov chain (MCMC). The vector of correlation parameters for this Gaussian process, \((\rho_{z,1}, \ldots, \rho_{z,p+m})\), will be denoted as \( \rho_z \) from here onwards.

The discrepancy term, \( \delta(\cdot) \), accounts for the systematic difference between the simulated and physical systems. The inadequacies in the computer code can come from the lack of understanding of the physics or approximations built into the computer code. Let \( \lambda_\delta \) denote the marginal precision of the discrepancy function and \( \rho_\delta = (\rho_{\delta,1}, \ldots, \rho_{\delta,p}) \) be the vector of correlation parameters for the \( p \) design variables. The GP model specified for the discrepancy term has a zero mean function and covariance function

\[
\text{Cov} \left[ Z(x), Z(x') \right] = \frac{1}{\lambda_\delta} \prod_{s=1}^{p} \rho_\delta^{4(x_s-x'_s)^2}. \tag{2.5}
\]

To complete the prior model formulation, the prior distributions for each of the marginal precision and correlation parameters governing the GPs have to be specified. Details of the
CHAPTER 2. BACKGROUND

prior distributions is deferred to later.

Let \( Y_f = (y_f(x_1), \ldots, y_f(x_n))^T \) be the vector of \( n \) observed field measurements and \( Y_c = (y_c(x_1, t_1), \ldots, y_c(x_{n^c}, t_{n^c}))^T \) be the vector of \( n^c \) simulations. The \((n + n^c)\)-vector \( Y_0^T = (Y_f^T, Y_c^T) \) is used to denote the vector of all available data. The first \( n \) components have associated inputs and calibration parameters \((x_1, \theta), \ldots, (x_n, \theta)\), whereas the remaining \( n^c \) components have inputs \((x_1, t_1), \ldots, (x_{n^c}, t_{n^c})\) that vary for each run of the computer code as specified by the experimental design.

The likelihood for \( Y_0 \) is

\[
L (Y_0|\theta, \mu, \lambda_z, \lambda_f, \rho_z, \rho_f) \propto |\Sigma_{Y_0}|^{-\frac{1}{2}} \exp \left\{ \left( Y_0 - \mu 1_{n+n^c} \right)^T \Sigma_{Y_0}^{-1} \left( Y_0 - \mu 1_{n+n^c} \right) \right\},
\]

(2.6)

where \( \mu \) is the constant mean and \( 1_d \) is a \( d \)-element column vector of 1. The covariance matrix of the likelihood is in the form of:

\[
\Sigma_{Y_0} = \Sigma_z + \begin{pmatrix}
\Sigma_{\delta} + \Sigma_y & 0_{n \times n^c} \\
0_{n^c \times n} & 0_{n^c \times n^c}
\end{pmatrix},
\]

(2.7)

where \( 0_{a \times b} \) is the \( a \times b \) matrix of zeroes. The matrix \( \Sigma_z \) is obtained by applying (2.4) to each pair of the \((n + n^c)\) observations and simulation outputs in \( Y_0 \). Equation (2.5) is applied only to each pair of field observations to construct the covariance matrix \( \Sigma_{\delta} \). Finally, the covariance matrix for the measurement error, \( \epsilon \), is given by the \( n \times n \) diagonal matrix \( \Sigma_y = (1/\lambda_y)I_n \), where \( I_n \) is an identity matrix of size \( n \).

2.1.2 Prior distributions and MCMC

The posterior distribution of the calibration and statistical model parameters \((\theta, \mu, \lambda_z, \lambda_\delta, \rho_z, \rho_\delta)\) is

\[
\pi (\theta, \mu, \lambda_z, \lambda_\delta, \rho_z, \rho_\delta|Y_0) \propto L (Y_0|\theta, \mu, \lambda_z, \lambda_\delta, \rho_z, \rho_\delta) \times \prod_{i=1}^{m} \pi (\theta_i)
\]

\[
\times \pi (\lambda_z) \times \pi (\lambda_\delta) \times \prod_{i=1}^{p+m} \pi (\rho_{z,i}) \times \prod_{i=1}^{p} \pi (\rho_{\delta,i}),
\]

(2.8)
where we abuse the notation and denote the prior distributions for $\theta$, $\rho_z$ and $\rho_\delta$ as

$$
\pi(\theta) = \prod_{i=1}^{m} \pi(\theta_i),
$$

(2.9)

$$
\pi(\rho_z) = \prod_{i=1}^{p+m} \pi(\rho_{z,i}),
$$

and

$$
\pi(\rho_\delta) = \prod_{i=1}^{p} \pi(\rho_{\delta,i}),
$$

respectively.

It is a common practice to scale all the measurements to have mean zero and variance 1, using the sample mean and standard deviation of the simulated outputs (Higdon et al., 2008; Linkletter et al., 2006). The normalization allows us to set $\mu = 0$, removes the prior distribution of $\mu$ from (2.8) and also minimizes the computing required to explore the posterior distribution of the parameters.

The calibration parameters are also scaled uniformly to the unit interval $[0, 1]$ (e.g., Linkletter et al., 2006). The prior for each of the components in $\theta$ is chosen to be an independent Normal distribution centered at 0.5 (center of the unit interval) with standard deviation 10. The choice of large standard deviation results in a weakly informative prior, allowing the data to move the calibration parameters.

The prior for the precision of the marginal variance, $\lambda_z$, is chosen to encourage its values to be close to 1 - the idea being that the computer model should capture much of the signal in the observations. We use a Gamma distribution (denoted generally as $\text{Gam}(a, b)$) for the prior for $\lambda_z$:

$$
\pi(\lambda_z) \propto \lambda_z^{a_z} \exp \{-b_z \lambda_z\}.
$$

(2.10)

When expert knowledge is unavailable, we have found that $a_z = b_z = 5$ works reasonably well as the choice centers the prior distribution at 1 with a reasonably large variance, thereby allowing for a fairly broad exploration of the posterior. Similarly, the priors chosen for the
remaining precision parameters are also Gamma distributions. We use the default prior distributions, \( \text{Gam}(1, 0.001) \), suggested by Higdon et al. (2004) for the hyperparameters of priors for \( \lambda_\delta \) and \( \lambda_y \). This specification implies a relatively uninformative prior for these precision parameters and encourages the data to choose a suitable value.

The components in \( \rho_z \) and \( \rho_\delta \) are bounded within the unit interval. Hence, a natural choice of prior for any \( \rho \in (\rho_z, \rho_\delta) \) is an independent Beta distribution (denoted \( \text{Beta}(c, d) \)) of the following form:

\[
\pi(\rho) \propto (\rho)^{c-1} (1 - \rho)^{d-1}.
\] (2.11)

Conventionally, the Beta priors are flat, with a mean near 1 and a small variance (e.g., Williams et al., 2006). This is based on the prior belief that all the inputs are equally uncorrelated to the simulator and allows the data to decide upon the dependence of the simulator on the different inputs by moving the \( \rho \)'s away from 1 in the posterior. In our experience the default choice of \( \text{Beta}(1, 0.001) \), suggested by Higdon et al. (2004) and Williams et al. (2006), encourages strong enough dependence in each of the parameters and works well in general.

The posterior density is commonly explored through a univariate random-walk Metropolis-Hastings updates (Metropolis et al., 1953; Hastings, 1970). Both types of updates work reasonably well in the examples that we have explored. Details of the MCMC such as the choice of proposal distributions and their widths will be discussed in the next few chapters.

2.1.3 Prediction at An Unknown Input Setting

An important goal of this endeavour is prediction. Using the posterior realizations of the statistical and calibration parameters from (2.8), we estimate the posterior predictive distribution for the new field measurement at the new input setting \( x^{\text{new}} \).

We begin by putting together the joint distribution of \( Y_0 \) and \( Y^f(x^{\text{new}}) \). Conditional on the parameters \( \theta, \lambda_z, \lambda_\delta, \rho_z \) and \( \rho_\delta \), the joint distribution is a multivariate Normal
distribution that has similar likelihood as $Y_0$:

$$
\begin{pmatrix}
Y_0 \\
Y^f(x_{new})
\end{pmatrix} | (\theta, \mu, \lambda_z, \lambda_\delta, \rho_z, \rho_\delta) \sim MVN(0, \Sigma^{new}).
$$

The joint distribution between $Y_0$ and $Y^f(x_{new})$ also has mean zero. The covariance matrix, $\Sigma^{new}$, is analogous to the covariance in (2.7) - there is an extra row and column in $\Sigma^{new}$ as a result of appending $Y^f(x_{new})$ to $Y_0$. The elements of the additional row and column is obtained by applying (2.4) and (2.5) to $x_{new}$ and the respective input settings of the data.

Through the usual properties of the multivariate normal distribution, the predictive distribution of $Y^f(x_{new})$, conditional on $Y_0$ and the parameters, is:

$$
Y^f(x_{new}) | (Y_0, \theta, \lambda_z, \lambda_\delta, \rho_z, \rho_\delta) \sim MVN(\mu_{pred}, \Sigma_{pred}),
$$

where $\mu_{pred} = \Sigma^{new}_{21} (\Sigma^{new}_{11})^{-1} Y_0$ and $\Sigma_{pred} = \Sigma^{new}_{22} - \Sigma^{new}_{21} (\Sigma^{new}_{11})^{-1} \Sigma^{new}_{12}$. The matrices $\Sigma^{new}_{ij}$ are sub-matrices of $\Sigma^{new}$ where

$$
\Sigma^{new} = \begin{pmatrix}
\Sigma^{new}_{11} & \Sigma^{new}_{12} \\
\Sigma^{new}_{21} & \Sigma^{new}_{22}
\end{pmatrix}.
$$

The sub-matrix $\Sigma^{new}_{11}$ is an $(n + n^c) \times (n + n^c)$ matrix whereas $\Sigma^{new}_{12}$ and $\Sigma^{new}_{21}$ are vectors of dimension $(n + n^c) \times 1$ and $1 \times (n + n^c)$, respectively. Lastly, the sub-matrix, $\Sigma^{new}_{22}$, is a scalar. The mean $\mu_{pred}$ is greatly influenced by both the observed and simulated measurements as it is a function of the data $Y_0$. The same can be say for the covariance of the predictive distribution of $Y^f(x_{new})$ since the $\Sigma^{new}_{11}$ is the same matrix in (2.7), which is the covariance matrix of the available data.

To make predictions, we first sample a vector of parameters from (2.8). Next, conditional on the sampled parameters, a prediction is sampled from (2.12). The sampling of parameters and predictions is repeated many times to provide estimated posterior quantities such as the posterior mean, variance and prediction intervals.
CHAPTER 2. BACKGROUND

2.2 Problems Addressed

In this thesis, the applications considered involve simulations from computer codes and actual observations of the physical system. In some applications, more than one computer code can be used to model the physical system. Some can be ranked based on their ability to model the field process, whereas the ranking criterion is unclear among some computer codes. New methodologies are developed in Chapter 3 – 5 to cater to these different scenarios. By combining the information from different computer codes, prediction and calibration can be improved.

Another important goal of computer experiments is to explore the features of the response surfaces of the field process. The global minimum and maximum of the process are usually of particular interest to the experimenters. In Chapter 6, we propose a modified expected improvement algorithm of Schonlau and Welch (1996) to identify the region of the global extremum in the field response surface. Through this algorithm, the design points that maximize information gain will be identified and added to the suitable computer code to aid the search of the global extremum.
Chapter 3

Multi-fidelity Model Calibration

In this chapter, a Bayesian hierarchical model that calibrates multi-fidelity computer simulators is proposed. Throughout, higher fidelity codes are assumed to better represent the real-world process but require more computing resources to simulate the system than lower fidelity codes. For ease of exposition and notation, we present the case where there are only two computer simulators – a high fidelity and a low fidelity model. It is conceptually easy to extend the proposed methodology to cases with more than two simulators, and this setting is discussed at the end of this chapter.

3.1 The Hierarchical Model

An important feature of the application that motivated the methodology in this chapter is that the calibration parameters for the computer models are not all the same. Some of the calibration parameters, $t_f = (t_{f,1}, \ldots, t_{f,m})$, are shared among the simulators, whereas others are required inputs only to individual computer models. The vectors of calibration inputs exclusive to the high and low fidelity models are denoted as $t_2 = (t_{2,1}, \ldots, t_{2,m_2})$ and $t_1 = (t_{1,1}, \ldots, t_{1,m_1})$, respectively.

First, consider the low fidelity computer model with inputs $(x, t_f, t_1)$ (i.e., the design variables and calibration parameters that are shared and unshared with the high fidelity
simulator). An output $y_1^c(\cdot)$ from the low fidelity simulator, $\eta_1^c(\cdot)$, is univariate and written as:

$$y_1^c(x, t_f, t_1) = \eta_1^c(x, t_f, t_1).$$

(3.1)

Similarly, the high fidelity simulator, $\eta_2^c(\cdot)$, has inputs $(x, t_f, t_2)$ and univariate output $y_2^c(\cdot)$:

$$y_2^c(x, t_f, t_2) = \eta_2^c(x, t_f, t_2).$$

Both simulators are used to describe the same process, but will not always give the same response. There are a few obvious reasons why this is the case. The lower fidelity model is inferior to the high fidelity simulator since it may, for example, fail to capture some processes that the high fidelity code can more accurately model. Furthermore, the two codes do not share all of the same inputs. The input vector $t_2$ only appears in the high fidelity model and thus, any impact that these variables have on the output cannot be captured by the low fidelity model. Similarly, the inputs $t_1$ appear only in the low fidelity model. To address these issues, we take the approach of writing the high fidelity simulator as a discrepancy adjusted version of the low fidelity model (e.g. Kennedy and O’Hagan, 2000; Qian et al., 2006):

$$y_2^c(x, t_f, t_2) = \eta_1^c(x, t_f, \theta_1) + \delta_2^c(x, t_f, t_2).$$

(3.2)

Specifying the first term in (3.2) as $\eta_1^c(x, t_f, \theta_1)$ amounts to partially calibrating (partially in the sense that the other calibration parameters must still be estimated) the first simulator to the second simulator. Recall, we will generally use $t$’s to denote inputs to the code and $\theta$’s to denote the optimal values for the calibration parameters, respectively. Furthermore, the discrepancy, $\delta_2^c(\cdot)$, represents the systematic differences between the partially calibrated low fidelity model and the high fidelity code. Lastly, notice that $\delta_2^c(\cdot)$ is a function of not only the design variables - as in Kennedy and O’Hagan (2001) - but also $(t_f, t_2)$. The calibration parameters are included in this discrepancy term because they can be modified in the high fidelity code. Therefore, unlike Kennedy and O’Hagan (2001) this discrepancy term captures the systematic differences in the outputs from the two computer models over values of the design variables, as well as the changes in the calibration inputs $t_f$ and $t_2$. 

In addition to simulator output, there are also field observations that are used to inform predictions. Since the higher fidelity simulator is assumed to better represent the physical process than the low fidelity simulator, it is natural to model the field observations with the computer simulator of highest fidelity. Similar to Kennedy and O’Hagan (2001), a discrepancy function, $\delta_f(\cdot)$, is used to capture the systematic inadequacy of the high fidelity simulator. The field observations are noisy versions of the mean process, and thus independent and identically distributed observational errors are included in our specification. For design variable setting, $x$, the univariate field process is written as:

$$y^f(x) = \eta_c^1(x, \theta_f, \theta_2) + \delta_f(x) + \epsilon,$$

where $\epsilon \sim N(0, 1/\lambda_y)$. Substituting (3.2) into (3.3) allows the field observations to be written:

$$y^f(x) = \eta_c^2(x, \theta_f, \theta_1) + \delta_2(x, \theta_f, \theta_2) + \delta_f(x) + \epsilon,$$

i.e. the response surface for the field data is written as the sum of the calibrated low fidelity simulator, the calibrated discrepancy between the two different simulators, the discrepancy between the high fidelity model and the data, as well as observational error. From here on out, we describe the response surfaces for the low and high fidelity simulators and the field data using the framework described in (3.1), (3.2) and (3.4), respectively.

It is possible at this point to envision applications with more than two simulators, each ranked from lowest to highest levels of fidelity. For example, consider the case where there are three simulators of different fidelity, $\eta_1^1(\cdot)$, $\eta_2^3(\cdot)$ and $\eta_3^3(\cdot)$, where $\eta_1^1(\cdot)$ is of the lowest fidelity and $\eta_3^3(\cdot)$ is best at describing the physical process. Outputs from $\eta_1^1(\cdot)$ and $\eta_2^3(\cdot)$ are modelled as above. Next, the relationship between $\eta_2^3(\cdot)$ and $\eta_3^3(\cdot)$ is similarly described by (3.1) and (3.2). Through substitution, $\eta_3^3(\cdot)$ can then be written as $\eta_1^1(\cdot)$ and two discrepancy functions. Finally, through the same sort of substitution, the field observations can be written as the sum of lowest fidelity simulator, a sequence of discrepancy terms and observational error. Readers are referred to Chapter 3.4 for more details.
3.1.1 Gaussian Process Models

The analysis of such experiments requires a probability model to describe the outputs. A reasonable choice is to estimate the response surfaces of the low fidelity simulator and discrepancies with independent GPs mentioned in Chapter 2.1.1.

We begin by first considering the specification for the low fidelity simulator. The outputs are treated as a realization of a random function as in (2.3). We follow the convention of most simulator applications by specifying the mean function as a constant, \( \mu \), and model the response surface through the covariance structure. The covariance between observations at inputs \((x, t_f, t_1)\) and \((x', t'_f, t'_1)\) is specified as

\[
\text{Cov} \left[ Z_{\eta_1}(x, t_f, t_1), Z_{\eta_1}(x', t'_f, t'_1) \right] = \frac{1}{\lambda_{\eta_1}} \prod_{s=1}^{p} \rho_{\eta_1,s}^2 \prod_{s=1}^{m_f} \rho_{\eta_1,p+s}^2 \prod_{s=1}^{m_1} \rho_{\eta_1,p+m_f+s}^2 ,
\]

where \( \lambda_{\eta_1} \) is the marginal precision of the GP for the low fidelity simulator. The \((p + m_f + m_1)\)-vector \( \rho_{\eta_1} \) is the vector of correlation parameters that govern the dependence in each of the component directions of \(x, t_f\) and \(t_1\) (e.g., Higdon et al., 2008; Linkletter et al., 2006).

The discrepancy, \( \delta_2(\cdot) \), captures the systematic differences between the high and low fidelity simulators as a function of the inputs, \((x, t_f, t_2)\), that are adjustable in the high fidelity model. Continuing as above, \( \delta_2(\cdot) \) is modelled as mean zero GP with covariance between simulations at inputs \((x, t_f, t_2)\) and \((x', t'_f, t'_2)\) specified as:

\[
\text{Cov} \left[ Z_{\delta_2}(x, t_f, t_2), Z_{\delta_2}(x', t'_f, t'_2) \right] = \frac{1}{\lambda_2} \prod_{s=1}^{p} \rho_{2,s}^2 \prod_{s=1}^{m_f} \rho_{2,p+s}^2 \prod_{s=1}^{m_1} \rho_{2,p+m_f+s}^2 ,
\]

where \( \lambda_2 \) is the marginal precision of the discrepancy function, and the vector of correlation parameters for this discrepancy function is \( \rho_2 \).

The discrepancy between the response from the high fidelity simulator and the mean of the physical process is also modelled with a zero mean GP as in (2.5). The marginal
precision and correlation parameters of this GP are denoted as $\lambda_f$ and $\rho_f = (\rho_{f,1}, \ldots, \rho_{f,p})$.

Denote the number of field observations, high fidelity and low fidelity simulation trials by $n$, $n_2^c$ and $n_1^c$ respectively. Furthermore, define the vector of all observations and simulation outputs as $Y = \left( Y_f^T, Y_2^c, Y_1^c \right)^T$, where $Y_f^T$ is as described in the previous chapter. The component $Y_2^c$ represents the $n_2^c \times 1$ vector of high fidelity simulator responses and $Y_1^c$ is the $n_1^c \times 1$ vector of low fidelity simulator outcomes. The field measurements have associated calibration parameters $(t_f, t_2, t_1) = (\theta_f, \theta_2, \theta_1)$. Similarly, the high fidelity simulator has associated calibration inputs $(t_f, t_2, \theta_1)$, where $(t_f, t_2)$ can vary for each run of high fidelity code specified by the experimental design.

To simplify notation, denote $\theta = (\theta_f, \theta_2, \theta_1)$, $\lambda = (\lambda_{\eta_1}, \lambda_2, \lambda_f)$ and $\rho = (\rho_{\eta_1}, \rho_2, \rho_f)$. The likelihood for $Y$ closely resembles (2.6):

$$L(Y|\theta, \mu, \lambda, \rho) \propto \Sigma_Y^{-\frac{1}{2}} \exp\left\{ \left( Y - \mu I_{n+n_2^c+n_1^c} \right)^T \Sigma_Y^{-1} \left( Y - \mu I_{n+n_2^c+n_1^c} \right) \right\},$$

but the covariance matrix $\Sigma_Y$ differs from (2.7) where

$$\Sigma_Y = \Sigma_{n_1} + \begin{pmatrix} \Sigma_2 & 0_{(n+n_2^c) \times n_1^c} \\ 0_{n_1^c \times (n+n_2^c)} & \Sigma_2 \end{pmatrix} + \begin{pmatrix} \Sigma_f + \Sigma_y & 0_{n_1^c \times (n+n_2^c)} \\ 0_{(n+n_2^c) \times n_1^c} & 0_{(n+n_2^c) \times (n+n_2^c)} \end{pmatrix}.$$  

The covariance matrix $\Sigma_{n_1}$ for the low fidelity simulator GP is obtained by applying (3.5) to each pair of the $(n + n_2^c + n_1^c)$-observations and simulation outputs in $Y$. Similarly, the covariance matrix $\Sigma_2$ is obtained by applying (3.6) to each pair of the $(n + n_2^c)$-observations and high fidelity simulator responses, $(Y_f^T, Y_2^c)^T$. The covariance matrix $\Sigma_f$ is obtained by applying (2.5) to each pair of field observations. The $n \times n$ covariance matrix for the measurement errors, $\Sigma_y$, is a diagonal matrix pointed out in the previous chapter.

### 3.1.2 Prior distributions and MCMC

The posterior distribution of calibration and statistical model parameters, $(\theta, \mu, \lambda, \rho)$, is similar to the posterior distribution (2.8). Through some rearrangements of the notation,
the posterior of the new model has the form

$$\pi(\theta, \mu, \lambda, \rho | Y) \propto L(Y|\theta, \mu, \lambda, \rho) \times \pi(\theta) \times \pi(\mu) \times \pi(\lambda) \times \pi(\rho).$$  

(3.9)

This posterior density function looks similar to (2.8) at first glance. However, the prior distribution for \(\theta = (\theta_f, \theta_2, \theta_1)\) here actually implies the product of all the independent prior distribution of \(\theta \in \theta\). The same can be said for the prior distributions of the correlation parameters where \(\pi(\rho)\) is the product of the prior distribution of \(\rho \in \rho = (\rho_{\eta}^1, \rho_2, \rho_f)\). To further simplify the notation, \(\pi(\lambda)\) is used to represent the prior for the marginal precision parameters of all three response surfaces, where

$$\pi(\lambda) = \pi(\lambda_{\eta}^1) \times \pi(\lambda_2) \times \pi(\lambda_f) \times \pi(\lambda_y).$$

The prior distributions of the parameters in this new model are very similar to the prior distributions of the model discussed in Chapter 2. We will briefly review the corresponding prior distributions for each parameter here.

As pointed out in the previous chapter, the prior distribution of \(\mu\) is unnecessary because the data is scaled to have mean zero and variance 1, using the sample mean and standard deviation of the simulated outputs. It is important to note that simulations from both the high and low fidelity computer models have to be pooled together to evaluate the sample mean and standard deviation.

The prior distribution of \(\lambda_z\) can be applied to the marginal precision parameter of the low fidelity simulator since \(\eta_1(\cdot)\) is expected to capture most of the signal from the field process. When experts’ knowledge is unavailable, the recommended default prior for \(\lambda_{\eta}^1\) is also \(\text{Gam}(5,5)\). Both the default prior distributions of the remaining precision parameters are chosen to be \(\text{Gam}(1,0.001)\). These default choices appear to work well in practice.

The calibration parameters are also scaled to the unit interval. The choice of prior is a weakly informative Normal distribution with mean 0.5 and standard deviation 10. The choice of prior distribution for each component of \(\rho\) continues to be the flat Beta prior,
Beta(1, 0.001) proposed in the previous chapter.

The posterior distribution for each parameter is explored using MCMC. Specifically, single-site Metropolis updates (Metropolis et al., 1953) are used for the components of $\rho$ and $\theta$. Proposals are made for each of these parameters from a uniform distribution centered at the parameter’s current value. The widths of the uniform distributions (one for each component parameter) are pre-computed by running short MCMC runs and choosing a width that gives an acceptance rate of about 0.44 (Gelman et al., 2004). Although this adjustment does not guarantee an acceptance probability of 0.44, we have found this procedure is helpful at choosing widths resulting in acceptance ratios between 0.25 and 0.75 and, more importantly, encourages the MCMC to converge faster. Good default choices for the widths for the updates can also be found using the method proposed by Graves (2011). For each of the precision parameters, we used Hastings updates (Hastings, 1970), where the proposed value is drawn from a uniform distribution centered at the current parameter value, with a width that is proportional to the parameter’s current value. We have found that a width that is 0.3 times the current parameter value (originally proposed by Higdon et al., 2008) works fairly well in general. It is feasible to use Metropolis updates for the precision parameters as well, but in the problems we have encountered, we have found the Hastings updates work just fine.

3.2 Prediction

The predictive distribution at unknown input settings for this model can be easily found by applying the technique discussed in Chapter 2.1.3. Here, we will briefly discuss the similarities and point out the differences in the predictive distribution $Yf(x^{new})$ at an unknown $x^{new}$ for problems mentioned in this and the previous chapter.

Conditional on $(\theta, \lambda, \rho)$, the joint distribution of the available outputs, $Y$, and an output at $Yf(x^{new})$, is also a multivariate normal distribution similar to (2.12). The covariance,
The predictive distribution of $Y_f(x^{new})$ is also obtained through the properties of the multivariate normal distribution mentioned in Chapter 2.1.3. However, the sub-matrix $\Sigma^{new}_{11}$ is an $(n + n^c_2 + n^c_1) \times (n + n^c_2 + n^c_1)$ matrix, which is the covariance matrix of the available data. Both sub-matrices $\Sigma^{new}_{12}$ and $\Sigma^{new}_{21}$ are of dimension $(n + n^c_2 + n^c_1) \times 1$ and $1 \times (n + n^c_2 + n^c_1)$. The sub-matrix $\Sigma_{22}$ remains a scalar in this case.

### 3.3 Examples

In this section, two examples are presented. The first example is a simple simulator that is used to demonstrate the proposed approach. After illustrating our implementation and some diagnostics to assess the adequacy of the model fit, a small simulation study is carried out to investigate the predictive performance of the proposed methodology. The second example is the application that motivated this work, and involves a radiative shock experiment conducted at CRASH. The main goal is to predict the observed field measurements given the outputs from two simulators and a set of field trials.

#### 3.3.1 Toy Example

We begin with the “toy” example in Bastos and O’Hagan (2009), with some slight alterations. That is, the setting has been modified to accommodate two simulators and field experiments. In addition, we refashion the computer models to include two design variables, a common calibration parameter and calibration parameters that exist in each simulator, respectively. For simplicity, all the input settings and calibration parameters are chosen from the unit interval.

We specify the low fidelity model as:

$$y_c^f(x, t_f, t_1) = \eta_c^f(x, t_f, t_1)$$

$$= \left(1 - \exp\left(-\frac{1}{2x_2}\right)\right) \frac{1000t_fx_1^3 + 1900x_1^2 + 2092x_1 + 60}{1000t_1x_1^3 + 500x_1^2 + 4x_1 + 20}.$$  \hspace{1cm} (3.10)
The high fidelity model is defined as the low fidelity response model plus a discrepancy term:

\[ y_2^f (x, t_f, t_2) = \eta_1^f (x, t_f, \theta_1) + \delta_2 (x, t_f, t_2), \]  

(3.11)

To illustrate the proposed approach, we simulate outputs from the respective models. Following Loeppky et al. (2009), we used a 40 run random Latin hypercube design (Mackay et al., 1979) for the low fidelity simulator. Since, in practice, the high fidelity model is likely to be more computationally expensive than the low fidelity model, only 10 runs are generated – also chosen using a random Latin hypercube design.

In most applications, there are relatively few field observations. Consequently to mimic this setting, only 3 field observations were simulated from the mathematical model:

\[ y^f (x) = \eta_1^f (x, \theta_f, \theta_1) + \delta_2 (x, \theta_f, \theta_2) + \frac{10x_1^2 + 4x_2^2}{50x_1x_2 + 10} + \epsilon \]

= \eta_1^f (x, \theta_f, \theta_1) + \delta_2 (x, \theta_f, \theta_2) + \delta_f (x) + \epsilon, \quad (3.12)

where \( \epsilon \sim N(0, 0.5^2) \).

For this example, the true value of the common calibration parameter is chosen to be \( \theta_f = 0.2 \), while the calibration parameter appearing only in the high and low fidelity models are chosen to be \( \theta_2 = 0.3 \) and \( \theta_1 = 0.1 \), respectively. These parameters are chosen to mimic the actual function proposed by Bastos and O’Hagan (2009).

Figure 3.1 displays the response surfaces for the two simulators and also the mean response surface for the field process. A quick glance at the figure reveals that the high fidelity model appears closer to the mean process than the low fidelity model. This represents the framework we are working within insofar as the high fidelity model is assumed to be more like the true system than the low fidelity model.

The posterior distribution of the model parameters was sampled using MCMC as outlined in Chapter 3.1.2. The MCMC chain is initialized with \( \theta_f = \theta_2 = \theta_1 = 0.5 \) (i.e., the
centre of the input space), $\lambda_{\eta} = 1$, $\lambda_2 = \lambda_f = \lambda_y = 20$ and all the correlation parameters, $\rho$ are chosen to be 0.1 as we assume that the simulator and discrepancies are dependent on all the inputs. Through some routine diagnostics such as the Geweke diagnostic (Geweke, 1992) and Raftery-Lewis diagnostic (Raftery and Lewis, 1992), we found that, for the data encountered in this example, different numbers of iterations (ranging from 2000 to 4000) are required for different parameters to achieve convergence. To be conservative, the MCMC was run for 10,000 steps, where the first 5,000 steps are treated as burn-in and discarded in further analysis.

In addition to the data simulated from (3.10) – (3.12) used to fit the proposed model (i.e., the training set), a validation dataset was generated from (3.12), so that the predictive performance can be evaluated. The validation set consisted of 25 field observations with input settings, $x$, chosen using random Latin hypercube sampling. We use the posterior mean prediction at $x$ to estimate $y^f(x)$. Figure 3.2 shows the predicted versus actual values for each of the validation points. The figure shows that the predictive model performs reasonably well since the points center around the $y = x$ line.

Figure 3.3 displays the deviations of the predictions from the true values plotted against
the predictions and also the input settings in each dimension. In each case, no obvious pattern is found in the plots, suggesting the outputs have similar degree of smoothness across the input space and that no obvious systematic behaviour was unaccounted for.

Figure 3.3: Diagnostics plots for the simple example: (a) Prediction error against predictions; (b) Prediction error against $x_1$; (c) Prediction error against $x_2$.

While not the specific goal of the proposed methodology, we now consider the estimation of the calibration parameters. Figure 3.4(a) shows the estimated one-dimensional and two-dimensional marginal posterior distributions of the calibration parameters. Vertical dashed lines are plotted at the true values of the calibration parameters. In general, these
posterior distributions can be interpreted as representing the uncertainty in the calibration parameters given the very limited number of observations and small numbers of simulations from imperfect simulators. A quick glance at the plots reveals that, except for $\theta_1$, the calibration parameters are not being constrained by the data. It is not too surprising that we can constrain $\theta_1$, but not the calibration of the other parameters, since there are more outputs (comparisons between the low and high fidelity models) to inform this parameter. The inability to constrain the other calibration parameters is likely due to the well known issue of confounding between the calibration parameters and the discrepancy functions (e.g., Loeppky et al. 2006) and the dearth of data. The confounding can be mitigated to some extent by the use of more informative prior distributions and also more observations.

Figures 3.4(b), 3.4(c) and 3.4(d) display the estimated posterior distributions of the calibration parameters for other sample sizes. The panels are the results of the simulations with (i) $n_1^c = n_2^c = 20$, $n = 3$, (ii) $n_1^c = n_2^c = n = 40$ and (iii) $n_1^c = n_2^c = n = 100$. The first case was chosen as a more simulation rich version of the above example. Comparing Figure 3.4(b) with the results in Figure 3.4(a), we see that the mode of the posterior distribution of $\theta_1$ is closer to the true value (dotted line) and there is less variability in the posterior distribution when there are more simulations. However, very little is learned about the calibration parameters $\theta_2$ and $\theta_f$. To gain more information on these parameters, there needs to be more field observations. Panels (c) and (d) consider cases where the number of simulations and field trials is larger than before. As the number of observations and simulations increases, the model is able to better estimate the calibration parameters. An interesting observation is that the shared calibration parameter $\theta_f$ is better constrained in panel (c) than $\theta_2$. The reason for this, we surmise, is that given the same number of field trials both the low and high fidelity models help inform $\theta_f$, but only the high fidelity model directly informs $\theta_2$. When there are relatively many simulations and observations, all of the calibration parameters tend to be well constrained (panel (d)).

A subsequent simulation study is performed to compare predictions of the new model
with approaches that only use some of the simulations. Models ML and MH are implementations of the Kennedy and O’Hagan (2001) approach using the data obtained from the low fidelity simulator and experiments, and outputs obtained from the high fidelity simulator and experiments, respectively. Predictions from these models are compared with those from the proposed approach which are denoted as MLH. In other words, we are investigating whether the proposed methodology of combining all simulations and observations is better in some sense than the Kennedy and O’Hagan (2001) method using one of either the low fidelity model or high fidelity model outputs alone.
Figure 3.5: Boxplots of the RMSPE obtained from the 100 simulated datasets analyzed using models ML, MH and MLH.

Figure 3.5 reveals that the RMSPE from the proposed model is consistently smaller than RMSPE of the other two models. Interestingly, in panel (a), we notice that the RMSPE is larger for the high fidelity model than the low fidelity model. This is the result of having relatively few runs of the high fidelity code. Looking at Figure 3.5(b), when $n_1^n = n_2^n = 20$
and \( n = 3 \), prediction using the higher fidelity outputs does better than prediction using only the low fidelity outputs. In either case, the proposed approach that uses all sources of data tends to do better in terms of RMSPE.

In general, we found that the proposed model that makes use of all the simulations works well for making predictions for the physical system. The simulation demonstrates that more efficient estimation is gained through this approach. Although calibration is not the priority, we come across a similar issue encountered by Kennedy and O’Hagan (2001) – calibration is difficult with limited amounts of data. However, as the number of outputs and observations increases, more information is available to calibrate the parameters of interest. In the case of calibration in our setting, it is important to note what is being achieved. That is, the posterior distributions reflect the uncertainty in the calibration parameters given the observations and the imperfect simulators.

3.3.2 CRASH

The application that motivated the proposed methodology arises from radiative shock experiments at CRASH. Figure 3.6 gives a diagram of the system that we want to predict. In the physical experiments, a high energy laser pulse irradiates a thin disk of beryllium at the front end of a xenon filled tube. The energy deposited in the surface causes the beryllium to ablate. A shock wave is then driven by the ablation pressure through the beryllium disk. After the shock wave breaks out of the beryllium disk, the disk acts as a piston, propagating the shock at a high speed into the xenon. When the xenon is shocked, it is heated to temperatures well over 100,000 °K and emits thermal x-ray radiation. These shocks are considered radiative when the radiation energy flux from the shock is high enough to impact the structure of the shock wave. Details regarding the radiative shock physics can be found in Drake et al. (2011). The radiating shock experiments that we are concerned with can be viewed as small-scale experiments for understanding astrophysical shock waves and other high temperature phenomena (McClarren et al. 2011; Drake et al. 2011).
Several measurements of interest are taken from each shock experiment and also simulations. We focus here on the time taken for the shock wave to exit the beryllium disk (breakout time). Our experiments were carried out at the OMEGA Laser Facility at the University of Rochester (Boehly et al. 1997). Two metrics were used to obtain measurements of the shock breakout. The streaked optical pyrometer (SOP) records a 2-dimensional image with optical light emission in one spatial dimension and is streaked in time in the other direction (Miller et al. 2007). To measure shock breakout, SOP views beryllium disk on the side opposite the laser irradiation. The detector sees no signal from the hot, shocked material until the shock reaches the back surface of the disk, indicating shock breakout. The other metric is the active shock breakout (ASBO) diagnostic (Barker and Hollenback 1972). ASBO reflects a probe laser off the back surface of the beryllium disk and utilizes interferometry for measuring the relative distance from the target to the diagnostic over time. When the shock reaches the back surface of the disk, it accelerates the surface, changing the distance from the detector to the target. Both the SOP and ASBO metrics can be used to provide measurements of the shock breakout time. In practice, we have used both to compute the breakout time for a shock and taken the average of the two inferred computed
values as the measurement. Of course, the shocks move very quickly (more than 100 km/s) and the breakout from the beryllium happens in a very short time period. For this set-up, the time measurement system measures the breakout time in pico-seconds ($10^{-12}$ s).

Using two different radiation-hydrodynamics codes (1D-CRASH and 2D-CRASH), we aim to predict the shock breakout time. The 2D-CRASH code includes two-dimensional processes and interactions that the one-dimensional code, or 1D-CRASH, does not. As a result, the 2D-CRASH model is assumed to be able to model the experiments better than the 1D-CRASH code, but it is also more computationally expensive.

The design variables for this experiment are the thickness of the beryllium disk ($x_1$) and laser energy ($x_2$). The electron flux limiter is calibration input to both simulators and is denoted as $t_f$. The laser energy scale factor is an additional calibration parameter, $t_1$, required by the 1D-CRASH code but not the 2D-CRASH simulator. The high fidelity computer code has two calibration inputs – beryllium gamma ($t_{2,1}$) and wall opacity scale factor ($t_{2,2}$). All the inputs are scaled to the unit interval before fitting the data to the proposed model.

We have 365 simulations from 1D-CRASH and 104 2D-CRASH runs available. The designs for each computer experiment were Latin hypercube designs, optimized using a space-filling criterion (Johnson et al. 1990). There are also 8 experiments that were conducted using the OMEGA laser where the breakout time was recorded.

The MCMC was set up as in the previous examples, with one exception. From previous usage of the laser, it was known that the observational standard error was about $50 \times 10^{-12}$ seconds - or approximately 1 after standardizing. A Gamma distribution with shape and scale parameter (10,000, 10,000) was chosen for the prior of $\lambda_y$. This is an informative prior that tightly centers the gamma distribution at 1. The widths for the Metropolis updates are
chosen as outlined in Section 3.1.2. The MCMC was run for 10,000 iterations. After performing the Geweke diagnostic (Geweke, 1992) and the Raftery-Lewis diagnostic (Raftery and Lewis, 1992), the first 3,000 iterations are treated as burn-in and discarded from the analysis.

Like the previous example, the deviations of the predictions from the observed breakout times are plotted against the predictions and the two input settings in Figure 3.7. No obvious pattern is found in any of the diagnostic plots, thereby suggesting that the model fit is adequate.

A leave-one-out study is conducted to evaluate the predictive ability of the new approach. That is, we delete an observation, fit the proposed model and predict the deleted observation. This is done for each of the 8 observations. Figure 3.8 is a plot of the resulting predictions against the observed breakout time. The 95% posterior prediction interval for each point is shown in the figure. The predictions are fairly close to the observed values and, thus most points are near to the $y = x$ line. However, the second observation from the left gives a prediction interval that almost fails to capture the observation.
Similar to the previous example, the proposed approach of combining all simulations and observations is compared to the Kennedy and O’Hagan (2001) method using one of either the 1D-CRASH or 2D-CRASH outputs alone. The results are shown in Figure 3.9. Looking at the figure, the proposed methodology has prediction intervals that are usually smaller than prediction intervals obtained from model ML and MH - though not universally so. The prediction intervals from model MH are generally the widest. The prediction intervals from the new methodology contained the observations, but the prediction of breakout time for shot (f) was outside of both prediction intervals from ML and MH.

The results in Figure 3.9 point to the proposed approach being generally more successful at predicting the breakout time than using the observations with the low or high fidelity simulators alone. With that said, the results are not as striking as in the previous section. The benefits of the proposed methodology will depend on issues related to the specific application. In the end, the quality of the predictions will be based on features such as the ability of the simulators to mimic the real process, the form of the discrepancies, and the number of simulations and observations. In this example, we had quite a few runs from both simulators and emulated both simulators fairly effectively - we could not know we could do
Figure 3.9: The horizontal dotted line in each subplot draws the actual observed breakout time. The first two bars of '+' on the far left of the plots are the simulated outputs from 1D and 2D-CRASH, respectively. The 95% prediction intervals, denoted as vertical intervals, are obtained from fitting models ML, MH and MLH, respectively. The mean of each prediction interval is denoted with circle.

so beforehand - but did not have very many observations.

Plots of the marginal posterior distributions of the calibration parameters are shown in Figure 3.10. The posterior distributions for all the calibration parameters, except the energy scale factor, are not constrained in this application. This is expected because of the limited number of experiments ($n = 8$) that inform these parameters.

3.4 Discussion

In this section, some extensions and limitations of the proposed approach are discussed while some avenues for future work are identified. So far, the focus has been on the setting where there are only two simulators. The new methodology, however, can easily be extended to
Figure 3.10: Plot of the two-dimensional marginals for the posterior distribution of the four parameters. The diagonals show the marginal posterior distributions of the calibration parameters. The off-diagonals sub-plots contain the two-dimensional marginal posterior distributions for the four calibration parameters. The solid lines represent the 95% high posterior density region.

model applications that involve more than two simulators. Suppose that there are $K$ simulators denoted as $\eta_k^c(\cdot)$ for $k = 1, \ldots, K$, where $\eta_k^c(\cdot)$ is the next highest level of fidelity model from $\eta_{k-1}^c(\cdot)$. The simulators share the same design variables, $x$, and some common calibration parameters, $t_f$. The remaining calibration parameters required by each of the respective simulators are denoted as $t_k$, for $k = 1, \ldots, K$. The intersection between $t_k$ and $t'_k (k \neq k')$ is empty and thus calibration parameters are included in either exactly one or all of the simulators.

In this work, the outputs from the computer simulator and field measurements are assumed to be univariate. The lowest fidelity simulator outputs is still denoted as (3.1). The outputs from the higher fidelity simulators can then be written as a combination of the lowest fidelity simulator and discrepancy functions that capture the systematic differences
between pairs of simulators. For \( k = 2, \ldots, K \), the simulated outputs are written:

\[
y^c_k(x, t_f, t_h) = y^c_k(x, t_f, t_k) = y^c_1(x, t_f, \theta_1) + \sum_{j=2}^{k-1} \delta_j(x, t_f, \theta_j) + \delta_k(x, t_f, t_k).
\]

The experimental observations are written as the sum of the low fidelity simulator and discrepancy functions:

\[
Y^f(x) = y^c_K(x, \theta_f, \theta_K) + \delta_f(x) + \epsilon = y^c_1(x, \theta_f, \theta_1) + \sum_{j=2}^{K} \delta_j(x, \theta_f, \theta_j) + \delta_f(x) + \epsilon,
\]

where \( \delta_f(x) \) measures the discrepancy between the highest fidelity simulator and physical process. The response surfaces of the different sources of data are modelled with GPs with mean and covariance functions discussed in Chapter 3.1.1.

While it is conceptually simple to extend the setting to more than two simulators, note that as the number of models grows, likely so too will the number of simulations. As a result, the covariance matrices (e.g., for the low fidelity model) can become so large that matrix inversion poses an additional computation problem. This issue occurs for all applications of GPs where many outputs are available. One way to deal with large data sets is to change the GP specifications and use a compactly supported covariance that reduces the computational effort through sparse matrix techniques (e.g., Kaufman et al. 2011). Alternatively, a multi-stage approach can be considered. That is, one would first emulate the lowest fidelity simulator using only outputs from that computer model. The next highest level of fidelity model is calibrated using outputs from this simulator and the lowest level of fidelity emulator. One can continue building models in a hierarchical fashion. This has the impact of reducing the size of the covariance matrices at each stage and thus the computational burden. A similar approach is used for implementing the model proposed by Kennedy and O’Hagan (2001) in Bayarri et al. (2007).
Our choice of sampling method from the posterior distribution – univariate random walk MCMC – has worked fairly well in the examples encountered. However, as the parameter space grows, a relatively large number of MCMC steps are required to allow the random walks to converge. In these cases, more efficient MCMC procedures are needed.

Some care should be taken in the prior specification for the precision parameters for the GPs. We have found that the default choices of prior distributions outlined in Section 3.1.1 work fine in most cases (e.g., the simulations in Section 3.3.1). However, for some datasets, extremely large values of $\lambda_y$ are observed. This amounts to essentially a model with no measurement error and discrepancies that are interpolating the noise. We noticed the phenomenon when the default priors are used for the CRASH example. This can also happen with the model proposed by Kennedy and O’Hagan (2001). In our case, we avoided this problem because we had a more informative prior distribution for $\lambda_y$. Alternatively, one can address this issue by rejecting small values of a precision parameter in the MCMC (this was done in Higdon et al. (2004)), or at the design stage by taking replicate field observations.

A further note of caution with respect to the experimental design is that the design regions for the computer experiments should coincide to avoid uncertainty due to extrapolation in the discrepancies between models. Suppose for example, the design for $t_f$ in the low fidelity simulator explores a much larger region than the design for the high fidelity model. When predictions are made, the proposed approach averages over the posterior distribution of the calibration parameters. For values of $\theta_f$ from the posterior that are outside of the range explored by the design of the high fidelity model, the proposed approach extrapolates $\delta_2(\cdot)$. This results in larger prediction intervals.

Lastly, there are some avenues for future research that can be envisioned in other applications. For example, the proposed model could be adapted to consider different output data-structures such as functional or spatial responses. This approach is not obviously suitable to settings where the simulators are not ranked by fidelity (e.g., climate models arising
from different research groups). In Chapter 5, a new methodology will be introduced for multiple computer codes of unknown fidelities. Further research can also be done for applications with more than two simulators that involve common calibration parameters that appear among a proper subset of the models or in settings with design variables that appear in only some, but not all, simulators.
Chapter 4

An Alternative Model for Model Calibration

In this chapter, a new Bayesian hierarchical framework for model calibration when there is a computer model and field data is introduced. The new model takes into account the common beliefs and structures of the simulated and physical systems in most computer experiments. A unique feature of the proposed model is the introduction of the bias term at the level of the computer code. This feature is a modification of the model proposed by Kennedy and O’Hagan (2001) that also sets the stage for modelling applications with multiple computer simulators, where no one model is believed to be better than another. In this chapter, the modelling framework is outlined for the setting where all the simulation runs from one computer simulator and some experimental observations are available. The independent GPs and their respective priors used to estimate the response surfaces of the systems will be discussed before the result of the calibration and prediction obtained through this methodology are illustrated at the end of this chapter.
4.1 Model Formulation

The applications considered in this chapter also have two types of data: (i) simulated outputs generated from a deterministic computer model, and (ii) field measurements obtained from experiments. There are usually fewer field measurements since running the computer model is cheaper than making direct observations of the physical process. A crucial belief in this new framework is that of the existence of a perfect simulator capable of capturing the physical process in a perfect manner. Such a simulator has been indirectly discussed in the literature such as Kennedy and O’Hagan (2001) and Higdon et al. (2008) and is modelled as the linear combination of the computer simulator and discrepancy function. The perfect simulator, denoted as $\zeta(\cdot)$, also relies on both the design variables and calibration parameters. The observation at $x$ is a noisy version of the perfect simulator with the calibration parameters at their true values, i.e.

$$y^f(x) = \zeta(x, \theta) + \epsilon. \quad (4.1)$$

Instead of modelling the observations as the linear combination of the computer code and discrepancy term in (2.2), a slightly different approach is taken here where the computer model is viewed as a biased version of the perfect simulator. Denote the bias term that captures the systematic difference as $\Delta(\cdot)$. The simulated response at input setting $(x, t)$ is a linear combination of $\zeta(\cdot)$ and the bias term, i.e.

$$y^c(x, t) = \eta^c(x, t) = \zeta(x, t) + \Delta(x). \quad (4.2)$$

The main focus in this chapter is to build a predictive model for applications with one computer simulator. From here on, the simulated responses are modelled as (4.2) while the field measurements are described with (4.1). For the remainder of this chapter, the basic setup and structure of this new model will be outlined in detail. The extension to the more interesting problem where there is more than one simulator will be described in Chapter 5.
4.1.1 Gaussian Processes

This new framework also requires the estimation of the response surfaces of the perfect simulator and the bias term for the purpose of data analysis and prediction. In a similar spirit to the work in Chapter 2, independent GPs will continue be deployed as the non-parametric regression estimator and basis for statistical inference for the outputs of deterministic function.

In the absence of measurement errors, the perfect simulator can be treated as a deterministic function and also, the system mean can be viewed as a realization of a random function in the form of (2.3). Denote the GP used to model this response surface as $Z_\zeta$. The mean of $Z_\zeta$ is the constant $\mu$ and has the covariance structure shown in (2.4). The parameters of this GP will have subscript $\zeta$ to differentiate them from the parameters for $Z$ in (2.4).

The bias term used to capture the inadequacy of the computer code, is modelled with a mean zero GP with the same covariance structure as in (2.5). The marginal precision of the bias term is denoted as $\lambda_\Delta$, whereas the vector of the $p$ correlation parameters is denoted as $\rho_\Delta$.

The available data is denoted as $Y_0 = \begin{pmatrix} Y^T, Y^cT \end{pmatrix}^T$. At first glance, the likelihood of $Y_0$, with mean $\mu 1_{n+n^c}$ and covariance matrix $\Sigma_{Y_0}'$, does not appear different from (3.7). However, it is important to note that this covariance matrix, $\Sigma_{Y_0}'$, has a slightly different structure from (2.7) since the systematic difference between the physical and simulated system is captured through the bias term modelled in the simulated outputs. The structure of this new covariance matrix is

$$\Sigma_{Y_0}' = \Sigma_\zeta + \begin{pmatrix} \Sigma_y & 0_{n \times n^c} \\ 0_{n^c \times n} & \Sigma_\Delta \end{pmatrix},$$

(4.3)

where the covariance matrix $\Sigma_\zeta'$ is obtained by applying (2.4) to each pair of the $(n + n^c)$ observations and simulated outputs in $Y_0$. By applying (2.5) to each pair of the simulated outputs, the covariance matrix $\Sigma_\Delta$ is formed. As in Chapter 2, $\Sigma_y$ is the covariance matrix for the measurement errors, which is a diagonal matrix of $\frac{1}{\lambda_y} I_n$. 
4.1.2 Prior distributions and MCMC

Both prior and posterior distributions of the calibration and statistical model parameters will be presented here. There is no change from the specification in Chapter 2 to the choice of default prior distribution of $\mu$, $\theta$, $\rho$, and $\rho_{\Delta}$. Readers are referred to Chapter 2 for details of these distributions. The focus here is the prior distributions for the marginal precision parameters.

As pointed out in the previous chapters, the data is scaled using the sample mean and standard deviation of the simulated outputs. The two main reasons for not using the sample mean and variance of the observations are (i) the number of observations is usually very limited and, (ii) the possibility of unusually large measurement error that can affect the estimated quantities. To accommodate our choice of normalization, the prior belief of $\lambda_\zeta$ and $\lambda_\Delta$ has to be chosen carefully. The perfect simulator is believed to capture most of the signals of the physical process. Hence, a suitable prior distribution for $\lambda_\zeta$ has mean close to 1. The marginal precision parameter of a perfect simulator smaller than 1 can be interpreted as the lack of signal from $\zeta(\cdot)$. Allowing the data to move the parameter around is important to understand the response surface. This can be achieved by choosing a large variance for the prior distribution of the marginal precision parameter. In the examples we have explored, the prior distribution $Gam(1, 1)$ has been found to work well in general.

Selecting a suitable prior distribution for the marginal precision parameter of the bias term is important. The magnitude of $\lambda_\Delta$ represents the significance of the inadequacy between the perfect simulator and the computer code. We found that the MCMC rejection rates for all the parameters are very high when the mean of the prior distribution for $\lambda_\Delta$ is too far from the acceptable values for the marginal precision parameter for the bias term. Hence, the starting value of $\lambda_\Delta$ in the MCMC is chosen to be the mean of the prior distribution and the hyperparameters for the prior distribution of $\lambda_\Delta$ are determined using the available data to encourage reasonable exploration of the posterior and faster MCMC convergence. There are other ways to determine the shape and scale parameters for the Gamma distribution.
Our default method relies on the average of the differences between each of the observation and their respective neighbouring simulations. The idea is that the systematic differences across the input space have similar magnitude. However, it is difficult to evaluate the actual discrepancy without knowing the true \( \theta \). It is also not always plausible to have observations and simulated runs at the same input, \( x \). Hence, the differences between the response surface of the simulated and physical system at \( x \) is approximated with the average of the distances between the observations and their respective neighbouring simulated outputs. This is done to get some useful help in setting up the prior distributions. Denote \( N(x_i) \) as a set of simulated outputs with inputs close to \( x_i \). We found that about 10% of the total simulations works generally well as the total amount of simulated outputs, i.e.

\[
N(x_i) = \{ y^c(x, t) \mid 0.1n^c x \text{ closest to } x_i \}.
\]  

Let \( D^2(x_i) \) be the mean squared difference between the observation \( y^f(x_i) \) and its neighbours in the set \( N(x_i) \). The mean of the prior distribution for the marginal variance of the bias term can be centered around the average of the mean squared differences, i.e. for \( \text{Gam}(a, b) \), \( \frac{a}{b} = 1/\bar{d} \), where

\[
\bar{d} = \frac{1}{n} \sum_{i=1}^{n} D^2(x_i).
\]  

To allow broader exploration of the posterior, we choose the variance of the Gamma distribution to be 1000 times \( 1/\bar{d} \). This decision, again, is application specific and can be adjusted based on prior knowledge. Based on our assumptions, the default prior distribution for \( \lambda_{\Delta} \) is \( \text{Gam}(0.001/\bar{d}, 0.001) \).

The posterior distributions for all the parameters are explored using single-site Metropolis updates (Metropolis et al., 1953). As pointed out in the previous chapter, the new proposed value comes from a Uniform distribution centered at the current value of the parameter. The Graves method (Graves, 2011) is used to compute the widths of all the Uniform distributions in the Metropolis updates. A few step widths (10 in our case) are selected for each parameter. Short MCMC runs are performed and acceptance ratios are
recorded. The acceptance ratios are then fitted to the pre-selected widths using a Generalized Linear Model (GLM) with a Binomial link function. The optimal width for each parameter is obtained by making a prediction at acceptance ratio 0.44 with the fitted GLM. In the event that the computed width is unreasonable (e.g. greater than 1 for the correlation parameters), some manual adjustment is required.

Prediction remains an important objective here. The joint distribution \( (Y_T^0, Y^f(x_{\text{new}}))^T \) has covariance structure similar to (4.3) and is estimated using the posterior realizations as specified in (2.8). The predictive distribution of \( Y^f(x_{\text{new}}) \) is evaluated in the similar method and the process of making prediction is as discussed in Chapter 2.

### 4.2 Examples

To illustrate the proposed model, three examples are considered. The first two, Examples 4.2.1 and 4.2.2, are based on a simple simulator. The implementation of the proposed methodology is illustrated on these examples, and some diagnostics are used to assess the model fit before investigating the calibration and predictive performance of the proposed methodology. In the last example, the new model will be used to predict the observations of the radiative shock experiment conducted at CRASH.

#### 4.2.1 A perfect computer model

The first two examples are adaptations of the “toy” example in Bastos and O’Hagan (2009). The setting is modified again to accommodate a computer simulator that contains a calibration parameter and some field observations. Similar to the example in Chapter 3.3.1, all the inputs and calibration parameters are chosen from the unit interval.

For this example, the perfect simulator, \( \zeta(\cdot) \), that governs the mean of the field process takes the form

\[
\zeta(x_1, x_2, \theta) = \left(1 - \exp\left(-\frac{1}{2x_2}\right)\right)\left(\frac{1000\theta x_1^3 + 1900x_2^2 + 2092x_1 + 60}{390x_1^3 + 500x_2^2 + 4x_1 + 20}\right).
\] (4.6)
In the next two examples, the true value of the calibration parameter is $\theta = 0.2$. To mimic a real situation where field measurements are hard to come by, only 5 observations are simulated from

$$y_f(x_1, x_2) = \zeta(x_1, x_2, \theta) + \epsilon,$$  \hspace{1cm} (4.7)

where the measurement error, $\epsilon$ comes from an independent Normal distribution with mean zero and standard deviation 0.2.

We begin by adding 30 simulations (Loeppky et al., 2006) from the computer model that has no systematic difference from the perfect simulator, i.e.

$$y_c(x_1, x_2, t) = \zeta(x_1, x_2, t).$$  \hspace{1cm} (4.8)

The inputs for the computer model in (4.8) come from a 30 runs random Latin hypercube design (Mackay et al., 1979).

We initialized the parameters the same way as in the example in Chapter 3.3.1. The calibration parameter is initialized in the center of the input space, i.e. $\theta = 0.5$. Based on the assumption of a strong signal from the perfect simulator, the marginal precision $\lambda_\zeta$ is started off at 1. Based on the data, the parameter $\lambda_\Delta$ is initialized at 1000 – the mean of $\text{Gam}(1,0.001)$. As pointed out in the previous chapter, all the correlation parameters are chosen to be 0.1 to represent our belief that the simulator and bias are dependent on all the inputs.

The MCMC was run for 22,000 steps, where the first 2,000 steps were used to compute the widths of the uniform distributions for the MCMC (Graves, 2011). The width for $\theta$ was adjusted to 0.2 because the proposed width was larger than 1 – an unreasonable width for $\theta$ that had been scaled to the unit interval. Routine diagnostics mentioned in the previous chapter are performed and the first 5,000 steps are discarded from further analysis.

After the training set has been fitted to the proposed model, a validation data set of size 25 is generated from (4.7) to evaluate the ability of the model to make good predictions. The
inputs of the validation data set are obtained through random Latin hypercube sampling. The median of the posterior predictions at $x$ is used to estimate $Y^f(x)$.

As shown in Figure 4.1, the points, which represent the predicted observations plotted against the actual values of all the validation points, centers around the line $y = x$. This indicates that the new predictive model performs quite well in this example.

Figure 4.2: Diagnostic plots: (a) Prediction error against predictions; (b) Prediction error against $x_1$; (c) Prediction error against $x_2$. 

Figure 4.1: Predicted versus actual observed outputs of the validation set. The straight line is $y = x$. 

CHAPTER 4. AN ALTERNATIVE MODEL FOR MODEL CALIBRATION
The model fit is also assessed with three diagnostic plots in Figure 4.2. The figure, which shows the deviation of the predictions from the true values, indicates no obvious pattern and suggests that most systematic behaviour had been accounted for. No obvious pattern is found in Figure 4.2 (b) and (c) as well. This convinces us that the outputs have similar degree of smoothness across $x_1$ and $x_2$.

![Figure 4.3: Marginal posterior distributions of the calibration parameters using the new model and commonly used model proposed by Kennedy and O’Hagan (2001).](image)

This example has been designed with no systematic difference between the physical and simulated systems. One of the reasons for doing so is to study the ability of the model to perform calibration and compare its performance with the calibration ability of the model proposed by Kennedy and O’Hagan (2001). The implementation of the model in Chapter 2, using the same set of simulations and observations will be referred to as model KO.

The posterior distribution of $\theta$ obtained through both models are shown in the top row of Figure 4.3. The true value of the calibration parameter is plotted as a vertical dashed line in each plot. Both posterior distributions do not show much difference from their prior uniform distributions. This is very likely due to the insufficient information coming from
the observed outputs. When the amount of observations is doubled, the calibration parameter can be better constrained, as shown in the bottom row of Figure 4.3. The new model appears to perform as well as the model proposed by Kennedy and O’Hagan (2001) qualitatively. The difference in the results can be largely due to Monte Carlo error and specification of the priors.

The predictive performance of the new model and model KO are compared in a simulation study. In this simulation study, 100 sets of training and validation data are generated independently through random Latin hypercube sampling. There are 30 simulations from (4.8) and 5 observed outputs from (4.7) in each training set. Each of the training sets is used to fit the proposed methodology and model KO. Predictions of the validation set are done and the RMSPE is used to evaluate these predictions. The results of the 100 training and validation sets are summarized in Figure 4.4. The similarity between the results from both models is not surprising. In fact, it is a promising indication that the proposed approach has similar predictive ability as the model proposed by Kennedy and O’Hagan (2001) in this simple setting.

![Figure 4.4: Boxplots of the RMSPE obtained from the 100 simulated datasets analyzed using the new model and the model proposed by Kennedy and O’Hagan (2001). Each dataset contains 30 outputs from the unbiased computer simulator and 5 observations.](image-url)
4.2.2 A more challenging example

The focus is now shifted to a variation of the computer model that has been discussed so far. A bias term will be added to the computer code, making the example more challenging. Model fit will be assessed, calibration and predictive performance will be evaluated in a manner similar to the earlier discussion.

Similar to the earlier example, only 5 observations will be generated from (4.7). However, the 30 simulated outputs are combinations of the simulator and bias terms, i.e.

\[ y_c(x_1, x_2, t) = \zeta(x_1, x_2, t) + \frac{x_1 x_2}{x_1^2 + x_2^2 + 1} + \eta(x_1, x_2, t) + \Delta(x_1, x_2). \] (4.9)

The initialization of the model parameters are the same as in the previous example and the posterior distribution of the model parameters was sampled using MCMC as outlined earlier as well. The first 1,000 iterations of the MCMC are used to determine the widths of the Uniform distribution in the Metropolis updates. The width for \( \theta \) proposed by the Graves method (Graves, 2011) also exceeded 1 in this example and hence, was adjusted to 0.2, the default step width of \( \theta \) in Chapter 3. Similar to the approach taken in the previous chapter, some MCMC convergence diagnostics are performed. The first 5,000 iterations are consider burn-in and discarded from future analysis as well.

The validation data set also contains 25 field observations with input settings, \((x_1, x_2)\), chosen using random Latin hypercube sampling. The posterior mean prediction at each of the \((x_1, x_2)\) are used to estimate \( Y_f(x_1, x_2) \). Figure 4.5 shows the predicted versus actual values for each of the validation points center around the \( y = x \) line.

The residuals are plotted against the predictions and also the input settings in each dimension. In Figure 4.6, no obvious pattern is found in the plots. This suggests that most systematic behaviour has been accounted for and the smoothness across both input spaces are of similar degree.
CHAPTER 4. AN ALTERNATIVE MODEL FOR MODEL CALIBRATION

Figure 4.5: Predicted versus actual observed outputs of the validation set. The straight line is $y = x$.

Figure 4.6: Diagnostic plots for this slightly realistic toy example: (a) Prediction error against predictions; (b) Prediction error against $x_1$; (c) Prediction error against $x_2$.

The results of parameter estimation from the new model and the model proposed by Kennedy and O’Hagan (2001) are compared. The top row of Figure 4.7 shows the estimated posterior distribution of $\theta$ of the two models. The true value of the calibration parameter is represented with vertical dashed lines. The estimation of the parameter becomes more accurate when the number of observations is increased to 10. It is convincing to conclude that the new model can perform as well as model KO but further comparison is done by evaluating the root mean squared differences between the posterior realizations of $\theta$ and the
actual calibration parameter in each case. The new model appears to be qualitatively the same as the model KO and again, the difference in the results can be largely assigned to Monte Carlo error and also the slight difference in prior specifications.

The subsequent simulation study in this example aims to evaluate the predictive performance of the new model. Similar to the previous simulation study, the results will be compared to the predictions obtained by the model proposed by Kennedy and O’Hagan (2001), which will continue be referred to as model KO. The simulation study involves 100 training and validation data sets that are generated independently through random Latin hypercube sampling. Each training set contains 30 simulated outputs and 5 observations. The new model and model KO are estimated for each training set and predictions of the validation set are obtained. The RMSPE for the validation data set is evaluated using the predictions. The results of the simulation study of the 100 training and validation data sets are summarized in Figure 4.8. Both boxplots appear to have similar medians and interquartile ranges. The RMSPEs obtained from the new model seem to be slightly larger than the RMSPEs evaluated under model KO at times. However, the difference is small enough that the performances of both models can be considered as equivalently good. This is anticipated as both frameworks are very much alike in the sense that the response surfaces of both the simulator and systematic inadequacy are modelled with GPs of the same structure.
In general, the new model did not outperform the model proposed by Kennedy and O’Hagan (2001) by much in the earlier simulation studies. The results of the simulation studies are comparable and share a lot of similarities in terms of their calibration and predictive performances. This is not surprising because the new model can be viewed as a modification of the one proposed by Kennedy and O’Hagan (2001), where the systematic discrepancy between the actual and simulated system has been moved down to the level of the computer simulator.

4.2.3 CRASH

The last example in this chapter aims to model the shock breakout time from the radiative shock experiments at CRASH. This is the same application that motivated the methodology in the previous chapter and readers are referred to Chapter 3.3.2 for the description of the experiment. The data set of interest is the 104 simulation runs from the 2D-CRASH code and the 8 observations.
As described earlier, the design variables of the 2D-CRASH code are the thickness of the beryllium disk and laser energy, which are denoted as $x_1$ and $x_2$. The three calibration inputs are the electron flux limiter ($t_1$), beryllium gamma ($t_2$) and wall opacity scale factor ($t_3$).

Before fitting the data to the new model, the inputs are scaled to the unit interval whereas the outputs are scaled to have mean zero and variance 1. The MCMC was set up as in the previous examples but with one modification – the prior for $\lambda_y$ was chosen to be $Gam(10,000,10,000)$. This setup is the same as the MCMC setup in the last example of Chapter 3.

The widths of the Uniform distribution in the Metropolis updates are determined with the Graves method (Graves, 2011) as pointed out in Chapter 4.1.2. We chose to sample the acceptance rate by running 200 MCMC iterations at each of the 10 different pre-chosen widths of the Uniform distribution in the MCMC for each parameter. The optimal widths proposed by this method aim to maintain the acceptance ratio at around 0.44 (Gelman et al., 2004). The step widths chosen for both $t_1$ and $t_3$ exceeded 1, and were manually set to 0.2 – the default step widths proposed by Higdon et al. (2008). The adjustment did not demonstrate any significant impact on the acceptance ratio of those parameters. Hence, the MCMC is run with the optimal step widths, with some minor alterations, for 20,000 iterations. We chose to treat the first 5,000 iterations as burn-in based on some routine MCMC convergence diagnostics. The analysis and predictions are done with the remaining sampled values.

The predictive ability of the new approach is evaluated through a series of leave-one-out studies. Similar to the leave-one-out study conducted in the example in Chapter 3.3.2, the new model is fitted with an observation removed and prediction is carried out for the deleted observed value. Figure 4.9 shows a plot of the predicted observations of this study against
the actual breakout time and the 95% posterior prediction intervals for each point. The points are reasonably centered around the $y = x$ line. All but one of the intervals capture the observed breakout times.

Figure 4.9: Predicted versus actual observed outputs of the validation set. The straight line is $y = x$.

The proposed approach is also compared to model MH. Recall, model MH is obtained by fitting the data to the model discussed in Chapter 2. A quick glance at Figure 4.10 shows that there is minimal difference between the results obtained though the new model and model MH. Both models fail to capture the same observed breakout time (i.e. shot (f)) and their 95% posterior prediction intervals are about the same length. However, at close inspection, we notice that the centers of the intervals from the new approach are generally closer to the observations (denoted with horizontal dashed line).

This result is consistent with our earlier conclusion that the new model has similar predictive ability as Kennedy and O'Hagan (2001), Higdon et al. (2004) and Higdon et al. (2008). In Figure 4.10, the marginal posterior distribution of the three calibration parameters are shown. The calibration results obtained through both models show a lot of similarities. This further convinces us that this model works as well as the model commonly used in the
Figure 4.10: The horizontal dotted line in each subplot draws the actual observed breakout time. The first bar of ‘+’ on the left of the plots are the simulated outputs from 2D-CRASH. The 95% prediction intervals, denoted as vertical intervals, are obtained from fitting models MH and the proposed model, respectively. The mean of each prediction interval is denoted with circle.

literature so far. It is not surprising to find that only the beryllium gamma is constrained. As mentioned in the previous chapter, improvement on both the prediction and calibration performance can be achieved by including the simulations from the 1D-CRASH code.

The results from the simulation studies and CRASH application show that the new model performs as well as the model proposed by Kennedy and O’Hagan (2001) when the simulated outputs come from one computer model. This is a promising finding as it allows us to further extend this new model for applications with multiple computer simulators, which will be discussed in the following chapter.
Figure 4.11: Plot of the two-dimensional marginals for the posterior distribution of the three parameters. The diagonals show the marginal posterior distributions of the calibration parameters. The off-diagonals sub-plots contain the two-dimensional marginal posterior distributions for the three calibration parameters. The solid lines represent the 95% high posterior density region.
Chapter 5

Computer Codes With No Ranking

In this chapter, a Bayesian hierarchical model that calibrates multiple computer simulators is proposed. Unlike Chapter 3, the computer codes are assumed to have similar fidelities or cannot be easily ranked by the experts. To ease explanation, only the case where there are only two computer simulators will be presented. The proposed methodology can easily be extended to cases with more than two simulators, and this setting is discussed at the end of the chapter.

5.1 The Bayesian Hierarchical Model

The problem considered in this chapter involves models that share the same design variables and calibration parameters. It is challenging to develop methodology for applications with different calibration parameters. Further discussion on this scenario can be found in Chapter 5.3 but will not be our main concern. The goal here is to build a predictive model for the field process while estimating the unknown calibration parameters.

The new methodology proposed in this chapter is an extension of the model presented in the previous section. The key difference is that each computer model will have its own bias function.
Suppose that there are two computer models. The first simulator can be written as the sum of the real system mean and a bias function:

\[
y_c^1(x, t) = \eta^1_c(x, t) = \zeta(x, t) + \Delta^1_c(x).
\]  

Suppose a second simulator is available that is not known to be better or worse than the first at representing the mean of the physical system. Let \( y_c^2(\cdot) \) denote the univariate output from the second computer simulator, \( \eta^2_c(\cdot) \). Using a different bias term \( \Delta^2_c(\cdot) \), the output can also be written as:

\[
y_c^2(x, t) = \eta^2_c(x, t) = \zeta(x, t) + \Delta^2_c(x).
\]

The two simulators are designed to mimic the same system, but may not give the same outputs at the same input settings. This is mainly due to the experts’ different understanding of the physical system (e.g. different climate model groups), the choice of numerical solvers used in the each of the computer codes and the overall implementations of the mathematics governing the physical system. These reasons often do not serve as a suitable ranking criterion for the computer simulators, making the model proposed in Chapter 3 unsuitable.

The response surfaces for the two simulators are written as the sum of the perfect simulator and the discrepancy between the \( \zeta(\cdot) \) and \( \eta^i_c(\cdot) \) for \( i = 1, 2 \). As before, we have noisy field observations that are represented by the sum of the perfect simulator and observation error. In this chapter, the response surfaces for the field measurement is described using (4.1) whereas the response surfaces for the two computer simulators are represented using the framework described in (5.2) and (5.3), respectively.

At this point, it is not hard to imagine an application with more than two computer codes. In the event that \( K \) computer simulators are anticipated, the outputs from \( k \)-th computer code, where \( k = 1, \ldots, K \), can be modelled as a linear combination of the perfect simulator \( \zeta(\cdot) \) and \( \Delta_k(\cdot) \). Detailed discussion will be presented at the end of the chapter.
5.1.1 Gaussian Processes

Again, GPs are used to model the response surfaces. We can use the framework outlined in Chapter 4. The only differences here is that with new computer model comes a new bias function.

The model described in this chapter is an extension of the model proposed in the previous chapter. It is natural to choose the constant mean GP with covariance structure (2.4) to model the perfect simulator. The response surfaces of the bias terms are also modelled in the same way as $\Delta(\cdot)$ in the previous chapter. Both GPs have zero mean but the covariance matrices have different parameter, i.e. the covariance between simulations of the $i$-th computer code at inputs $\mathbf{x}$ and $\mathbf{x}'$ is

$$
\text{Cov} \left[Z_{\Delta_i}(\mathbf{x}), Z_{\Delta_i}(\mathbf{x}')\right] = \frac{1}{\lambda_{\Delta_i}} \prod_{s=1}^{p} \rho_{\Delta_i}^4 (x_s - x'_s)^2.
$$

(5.3)

The likelihood of $\mathbf{Y} = \left(\mathbf{Y}_f^T, \mathbf{Y}_2^c, \mathbf{Y}_1^c\right)^T$ is similar to (3.7). It also has a vector of constant mean, $\mu \mathbf{1}_{n+\hat{n}_1+\hat{n}_2}$, but the covariance matrix, $\Sigma'_{\mathbf{Y}}$ closely resembles (4.3), where

$$
\Sigma'_{\mathbf{Y}} = \Sigma_{\zeta} + \begin{pmatrix}
\Sigma_y & 0_{n \times \hat{n}_1} & 0_{n \times \hat{n}_2} \\
0_{\hat{n}_1 \times n} & \Sigma_1 & 0_{\hat{n}_1 \times \hat{n}_2} \\
0_{\hat{n}_2 \times n} & 0_{\hat{n}_2 \times \hat{n}_1} & \Sigma_2
\end{pmatrix},
$$

(5.4)

The first term on the right hand side of (5.4) is formed by applying (2.4) to each pair of observations and simulations. The covariance between each pair of the simulated outputs from the first simulator is evaluated using (5.3) to create $\Sigma_1$, and $\Sigma_2$ is obtained in the same way.

The same prior distributions of the calibration and statistical model parameters that were specified in Chapter 4 will be used here. The same can be said for the posterior density of the parameters, $\pi(\theta, \mu, \lambda, \rho | \mathbf{Y})$ and the single-site Metropolis update used in this
framework. The predictive distribution and prediction of $Y_f(x^{new})$ is also a straightforward extension of the procedure discussed in Chapters 2 and 3.

## 5.2 Examples

In this section, two examples are used to illustrate the proposed methodology. The perfect simulator in both examples is a modification of the Goldstein and Price function that has been widely used in the literature of global optimization (Price, 1983; Chelouah and Siarry, 2003; Fan et al., 2004). The implementation of each example is illustrated and some diagnostics are carried out to assess the adequacy of the model fit before a few small simulation studies are performed to investigate the predictive performance of the proposed methodology.

The Goldstein and Price function is altered to accommodate a calibration parameter, where

$$\zeta(x, t) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))$$

$$\times (30 + (2x_1 - 3x_2)^2(18 - 80tx_1 + 30tx_1^2 + 120tx_2 - 90tx_1x_2 + 27x_2^2))$$

(5.5)

The calibration parameters are chosen from the unit interval, while the input settings take the values between $-2$ and $2$ as in the actual Goldstein and Price function. The true calibration parameter, $\theta$, is chosen to be 0.4. For the field observations, white noise, $\epsilon \sim N(0, 0.5^2)$ is added to every observation.

### 5.2.1 A Realistic Example

This example is used to mimic real applications involving two computer models that are about the same or unknown fidelity. Two different bias terms are added to the perfect simulator, $\eta(\cdot)$, to create the two computer models for this example.

Let $\Delta_1(\cdot)$ denote the bias term for the first computer model. The output from the first
computer model is a linear combination of the perfect simulator where:

\[
y_{c1}(\mathbf{x}, t) = \zeta(\mathbf{x}, t) + (40 - 56x_1 + 72x_1^2 + 65x_2 - 25x_2^2 - 72x_1x_2) (20 + 27 \exp \{x_1x_2\})
\]

\[
= \zeta(\mathbf{x}, t) + \Delta_1(\mathbf{x}). \tag{5.6}
\]

The second computer model is built in the similar way using a different bias term denoted as \(\Delta_2(\cdot)\). Our choice of bias term for the second computer code is the negative of \(\Delta_1(\cdot)\), i.e. \(\Delta_2(\mathbf{x}) = -\Delta_1(\mathbf{x})\) and the second computer code is given by

\[
y_{c2}(\mathbf{x}, t) = \zeta(\mathbf{x}, t) + \Delta_2(\mathbf{x}). \tag{5.7}
\]

Figure 5.1 shows the response surfaces of the system created for this example. Both response surfaces of the computer simulators (shown in the middle and far right panels of Figure 5.1) appear to be similar to the mean response surface of the Goldstein and Price function. This represents the motivating framework of interest where it is difficult to tell apart the better computer model.

Figure 5.1: Surfaces of the Goldstein and Price function and the two computer codes.

As described in Loepkky et al. (2010), since the computer models are functions of three inputs (including tuning parameter), 30 outputs are simulated from each computer model based on independent Latin hypercubes. However, only 5 experimental observations are generated from (5.5) to mirror the minimal amount of field measurements in many real applications.
The step width of the Uniform distribution for each parameter is also determined using Graves method (Graves, 2011) before examining the posterior distribution of the model parameters. Based on the adjusted step sizes, the posterior distribution of the model parameters is sampled using single-site MCMC. Since the MCMC achieves convergence well before 20,000 steps, the MCMC is stopped after 20,000 steps and the first 3,000 steps are removed before performing further analysis.

As usual, the model fit has to be assessed before further analysis and studies can be done. The validity of the model is verified by analyzing the plots of prediction errors versus the true values and input settings. Predictions are made on a hold-out set of 25 input settings chosen via Latin hypercube sampling (Mackay et al., 1979). Predictions for the field measurements are performed at these inputs and the relevant diagnostic plots are shown in Figure 5.2. There is no obvious pattern in these plots, suggesting no evident indication of bad model fitting.

![Figure 5.2](image)

Figure 5.2: Error of prediction versus prediction and input settings.

In order to compare the predictive ability, a simulation study is carried out by comparing the predictions obtained through the new methodology with approaches that only use some of the simulations (Kennedy and O’Hagan, 2001; Higdon et al., 2004). The objective of this simulation study is to investigate the effect of using data from the two models on
the predictive performance, in comparison to the predictions obtained through the model fit of Kennedy and O’Hagan (2001) using simulations from only one computer simulator. Model M1 is the implementation of the model in Chapter 2 using only simulated outputs from the first computer model, while model M2 uses only data from the second computer model. The model fit using the new methodology is denoted as M.

The simulation study involves 100 independent training and validation sets. Each training set consists of 5 observations obtained from (5.5) and 30 simulations from (5.6) and (5.7). First, each training set is fit using M, M1 and M2. The predictions of the validation set are obtained for each model fit and the RMSPEs are evaluated. The average length of the 95% credible intervals for each validation set is also recorded.

Figure 5.3 shows the RMSPEs of the models. The boxplot on the far left shows that predictions from the proposed model (M) have RMSPEs that are consistently smaller than the RMSPEs of models M1 and M2. This shows that predictions obtained from the new methodology are better in the sense that most of the predicted field measurements are closer to the true values.

![Figure 5.3: RMSPE of the simulation study.](image)
Another aspect of the prediction considered here is the length of the credible intervals. Although large credible intervals will likely contain the true value more often, they are generally not as informative. In Figure 5.4, the boxplots show that model M is usually more capable of consistently producing smaller prediction intervals than the individual models alone. The credible intervals captured through models M1 and M2 are wider and appear to contain more uncertainty than those from the new methodology. It has not gone unnoticed that the results (shown in Figure 5.3 and 5.4) obtained through both M1 and M2 are similar. We expect this since the simulated data used in both models come from computer simulators of the similar fidelities.

![Figure 5.4: Average length of the 95% credible intervals obtained from the simulation study.](image)

The calibration performance of the new methodology is also compared to the performance of model the Kennedy and O’Hagan model for each individual simulator (M1 and M2). In this comparison, two other models, denoted as M1* and M2*, are included as well. Both models are implementation of the model proposed by Kennedy and O’Hagan (2001). Similar to model M1, the model M1* uses simulated responses from the first computer simulator but the number of simulations is increased to 60 – the total amount of simulated outputs used in model M. Similarly, model M2* uses 60 simulation runs from 5.7. Figure 5.5 presents the posterior distributions of the calibration parameters obtained through the
different models. The main reason for having the new models, which have more simulations, is to demonstrate that run-size does not explain all of the difference in our results. When there are 5 observations, none of the models do terribly well at calibration. That is, none of the posterior distributions of the calibration parameter are centred at the true value (shown as vertical lines). However, in the bottom row of Figure 5.5, the calibration parameter appears to be better constrained in models M, M1* and M2* when the number of field measurements is increased to 10. The calibration performance has shown significant improvement due to the increase in the number of simulation runs. Hence, it is in our favour to include as many simulated outputs as possible to inform the model for the purpose of improving the calibration performance. This is more easily attainable through the new model that takes into account all available simulations from different computer codes.

Figure 5.5: Posterior distributions of $\theta$ obtained from different models. The first row is the results of models with 5 observations while the second row shows the posterior distributions of models with 10 field measurements. The vertical line is $\theta = 0.4$, the actual value of the calibration parameter.
5.2.2 Computer Simulators When One Model is Better Than Another

In some applications, one computer model may be better than another, but this fact is unknown to the scientists. With this in mind, the second example is geared towards understanding the prediction and calibration performance of the proposed approach when, unbeknownst to the investigators, one simulator is superior to the other. The implementation, model and predictive assessment of this example bears resemblance to the previous examples.

Observations and simulated outputs from the first computer model are generated as in the previous example, i.e. the observed data is obtained from (5.5) and the simulated data from the first computer model comes from (5.6). The second computer simulator, however, is slightly modified. The outputs are generated from $\zeta(\cdot)$ alone, making the second computer model exactly the same as the modified Goldstein and Price function without observational error, except that the calibration parameter can be adjusted in the computer code. The missing bias term in the second computer simulator creates the difference in quality (i.e., one is systematically biased and the other is not) among the two computer codes.

Following the earlier example, 5 observations and 30 simulated outputs are available for the training set. Assuming no knowledge of the difference in fidelity, the data set is fitted using the proposed methodology. The intention is to investigate the predictions and calibration performance of the new model when the data originates from multiple models where one is better than the other.

Similar to the earlier examples, the model fit is investigated using diagnostic plots involving the prediction of validation set consisting 25 input settings $x$ sample randomly using Latin hypercube sampling (Mackay et al., 1979). The diagnostic plots are summarized in Figure 5.6. There appears to be no obvious pattern in any of the plots, which suggests no evident of bad model fitting of this data using the new model.
A subsequent simulation study is performed to compare the predictive ability of the new model and approach proposed by Kennedy and O’Hagan (2001) that do not use all available simulations. This simulation study is a replication of the earlier study performed in the earlier examples. In this example, the model fit using the new methodology is denoted as $M^+$. The reason for using $M^+$ instead of $M$ is to avoid confusion with the model fit in the previous example. Model $M_3$ fits the data from the first computer code and the field measurements using the model mentioned in Chapter 2. The same applies to model $M_4$ that uses the data from the second computer simulator and field observations.

The predictive ability is also compared using the RMSPE and the average length of the 95% credible intervals. Figure 5.7 shows that the RMSPEs obtained from the proposed model are usually the smallest among the 100 simulation sets.

Figure 5.8 shows that the 95% credible intervals obtained from the proposed methodology are also shortest among the models, implying more informative and reliable predictions.

An attempt is also made to compare the calibration in this example. Similar to the approach in the example in Section 5.2.1, two models $M_3^*$ and $M_4^*$ are added as well. Both of these models involved 60 simulations each and are calibrated with the methodology that
only use outputs from one computer code. The first row of Figure 5.9 shows that all models are unable to calibrate $\theta$ when there are 5 observations. However, when the number of field measurements is increased to 10, the posterior distributions of $\theta$ under all models, except model M3, are centered closer to the true value, as shown in the bottom left panel of Figure 5.9. It is not surprising to find both models M4 and $M_4^*$ can identify $\theta$ as the number of field measurements increases. This is because of the lack of systematic difference in the simulated outputs used to inform the models. The existence of bias in the simulated outputs affects how well the calibration can be done. This is obvious in the lack of success shown in the posterior distributions obtained from model M3. The model does not appear to be affected by the increase in the number of field measurements. Although some of the simulated outputs used to inform model $M^+$ are biased, the new methodology can determine $\theta$ as well as models M4 and $M_4^*$. In the event of computer codes of unknown fidelity, combining outputs from all sources has proven to be beneficial in terms of calibration as well.

Based on the comparisons in both examples, we found that the proposed model that makes use of all the simulations can make more informative predictions for the physical system. In both circumstances, we found that calibration is difficult when there is limited
amount of data. The calibration effort using the new methodology has also shown more success as well when the number of observations increases. In short, this new methodology is not only suitable for applications with multiple computer codes with similar fidelities, but also applicable to applications with unknown fidelities.

5.3 Discussion

So far, the focus has been on the setting where there are only two computer codes. The new methodology, however, can be easily extended to model applications that involve more than two computer codes. Assume that there are $K$ independent computer codes of similar or unknown fidelities, denoted as $\eta^c_k(\cdot)$ for $k = 1, \ldots, K$. For $k = 1, \ldots, K$, a univariate output from the $k$-th simulator, $y^c_k(\cdot)$ is then given by

$$y^c_k(x, t) = \eta^c_k(x, t).$$

(5.8)

As shown in earlier, the output can be represented as a summation of the perfect simulator and an independent bias term, $\Delta_k(\cdot)$, such that

$$y^c_k(x, t) = \zeta(x, t) + \Delta_k(x),$$

(5.9)
CHAPTER 5. COMPUTER CODES WITH NO RANKING

Figure 5.9: Posterior distributions of $\theta$ obtained from different models. The first row is the results of models with 5 observations while the second row shows the posterior distributions of models with 10 field measurements. The vertical line is the actual value of the calibration parameter.

where $\zeta(\cdot)$ is the perfect simulator. The response surfaces of the perfect simulator and the bias terms are modelled as discussed in 5.1.1. The GPs of the different bias terms are differentiated by having distinct subscripts.
Chapter 6

Sequential Kriging Optimization

The expected improvement (EI) algorithm (Schonlau and Welch, 1996) is a sequential design strategy for finding global extrema of a deterministic function. Huang et al. (2006a) extended the algorithm to utilize outputs from multi-fidelity computer simulators to identify the global extremum of the system with the highest fidelity. The applications considered in the literature do not involve field measurements and calibration parameters. The innovation in this chapter is to develop EI methodology to find optima in the setting where there are multiple computer codes - with possible different levels of fidelity - for model calibration problems. For ease of exposition and notation, we present the case where there are only two computer simulators. The computer models can have different fidelities – high fidelity and low fidelity models – or have similar fidelities. Extending the proposed methodology to cases with more than two simulators is straightforward, and this setting is briefly discussed at the end of this chapter. The new algorithm will be illustrated with a few examples similar to those discussed in Chapters 3 and 5.

6.1 The Improvement Function

The feature of interest here will be the minimum of the real system. In the case where the maximum of the real system is of interest, the function can be multiplied by $-1$ and minimized.
More formally, within a feasible region, $\chi$, our interest is in

$$x_{min} = \arg\min_{x \in \chi} y_f(x).$$

A naive approach for estimating $x_{min}$ is to simply perform an $n$-trial experiment design, use the model calibration methodology, optimize the estimated response surface for the field data and then estimate $x_{min}$. Such an approach is likely to be inefficient because it is likely to concentrate the sampling efforts in regions of the input space that provide little help in estimating the location of the minimum.

The challenge is to efficiently use the available resources to estimate the location of the minimum. The existing algorithms for multi-fidelity codes (e.g. Huang et al. (2006a), Huang et al. (2006b) and Forrester et al. (2007)) propose new inputs and also the most suitable computer model to perform the simulation together using a single criterion. These approaches are not suitable in multi-fidelity calibration problems. Those algorithms also rely heavily on the information from the computer simulators resulting in inaccurate estimation of $x_{min}$ or slow convergence. We attempt to address these issues by introducing a two-stage approach, where the new input, $x^+$ and the choice of computer simulator to perform the new run are selected using two separate criteria.

In a spirit similar to the work of Schonlau and Welch (1996) and Jones et al. (1998), $x_{min}$ can be identified in an efficient manner by maximizing the information coming from the new computer trials. At any unsampled point $x$, the mean of $y^f(x)$ is unknown. The uncertainty of $y^f(x)$ can be modelled by treating the expected response surface of $y^f(x)$ as a realization from a Normal distribution with mean $\mu^f(x)$ and standard deviation $s(x)$. Define the improvement function as

$$I(x) = \max(y_{min}^f - y^f(x), 0), \quad (6.1)$$

where $y_{min}^f$ denotes the smallest field measurement observed by the experimenter. The
improvement is then averaged over $y_f(x)$, giving the expected improvement

$$E[I(x)] = \begin{cases} 
(y_{min}^f - \mu_f(x)) \Phi\left(\frac{y_{min}^f - \mu_f(x)}{s(x)}\right) + s(x) \phi\left(\frac{y_{min}^f - \mu_f(x)}{s(x)}\right), & s(x) \neq 0 \\
0, & s(x) = 0 
\end{cases} \tag{6.2}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the standard Normal probability density function (pdf) and cumulative distribution function (cdf).

The expression for $E[I(x)]$ contains two parts. The first term tends to be large at points where the field observations have small predicted values, whereas the second term is large when the uncertainty of $y_f(x)$ is large. Thus, the criterion encourages sampling at values near potential minima and also in unsampled regions where the minimum may lie. The EI is the criterion for the first stage of the proposed approach. The design variable $x$ that maximizes (6.2) is of interest.

The second stage of the proposed methodology focuses on the selection of computer simulator for evaluating the input that maximizes the EI in the first stage. A key component in the criterion that is used to select the most suitable computer code is the correlation between the predictive distributions of the field process and computer simulator. Computer models that can better capture the physical system will have a large correlation. Define the correlation between the true and simulated response surface at $x$ as

$$corr\left[Y_f^r(x), Y_i^c(x, t)\right]$$

The proposed criterion for selecting the more efficient computer simulator is a combination of the improvement function, correlation between the response surface and the cost that is required for each system:

$$C(i|x) = E[I(x, i)] \alpha_1(x, i) \alpha_2(i). \tag{6.3}$$

We begin by examining the EI term on the right hand side of the equation and defer the discussion of $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ to later. The first term in the criterion has the a similar form
as (6.2), i.e.

\[
E [I(x, i)] = \begin{cases} 
(y_{\min} - \mu_i^c(x, t)) \Phi \left( \frac{y_{\min} - \mu_i^c(x, t)}{s_i^c(x, t)} \right) + s_i^c(x, t) \phi \left( \frac{y_{\min} - \mu_i^c(x, t)}{s_i^c(x, t)} \right), & s_i^c(x, t) \neq 0 \\
0, & s_i^c(x, t) = 0
\end{cases}
\] (6.4)

where \(\mu_i^c(x, t)\) and \(s_i^c(x, t)\) are the mean and standard deviation of \(Y_i^c(x, t)\). Here, the notation is somewhat abused where \(t\) is the vector of all calibration parameters required by the \(i\)-th computer simulator. Recall that the computer simulators are dependent on calibration parameters that can be shared or unshared among the different computer codes. The second term in (6.3) is the correlation between the predictive distributions of the computer simulator and the actual system at \(x\). It is used to account for the reduction in reward when a lower fidelity computer simulator is used.

The term \(\alpha_2(i)\) is the ratio between the cost-per-evaluation on the physical system and that on the \(i\)-th computer code, i.e.

\[
\alpha_2(i) = \frac{C_f}{C_i^c},
\]

where \(C_f\) is the cost of performing an experimental run and \(C_i^c\) is the cost of evaluating the computer model. The cost can be the computing resources, time or financial expense required to execute the experiment or the computer code. This ratio accounts for the evaluation costs and further adjusts the sampling strategy. Performing field experiment is more costly in most applications. In the event that one computer model has higher fidelity than the other, the higher fidelity computer model is usually more expensive to evaluate, i.e. \(C_2^c > C_1^c\). On the other hand, when both computer models have similar fidelity, the cost-per-evaluation of the simulators are close to identical and \(C_2^c \approx C_1^c\). Without loss of generality, our assumption is that \(C_f > C_2^c \geq C_1^c\).

The criterion (6.3) is the augmented EI function proposed by Huang et al. (2006a) to select both \(x^+\) and the computer code used to evaluate the new input at the same time. The criterion works well for deterministic functions with no calibration parameter as the
response surfaces remain the same throughout. In our setting, however, the estimated calibration parameter and response surfaces change as new simulation runs are added to the system. To minimize the effect of the simulated response surface has on the search for $x_{min}$, the decision on the new input is made solely based on the estimated field surface. The computer model selected to perform the new simulation run is decided by the cost and information gain. In short, the two stages in the proposed methodology are:

**Stage 1.** The new input has the greatest expected improvement in (6.2).

**Stage 2.** The choice of computer code used to evaluate $x$ from Stage 1 is determined using $C(i|x)$.

### 6.2 The Implementation

The proposed approach is sequential in nature and involves optimizing (6.2) and (6.3) at each step. We will first address the issue of computing $\mu_i^c(x, t)$, $s_i^c(x, t)$ and $\alpha(x, i)$, and then move on to explain the sequential design strategy.

When performing the computer simulator at the design point $x$, the calibration parameter has to be specified as well since the computer models are dependent on both $x$ and $t$. At this point, the true calibration parameters are still unknown and in principle, the prediction of $y_i^c(x, t)$ has to be averaged over the posterior distribution of $\theta$. However, it can be computationally intensive to average over the posterior and the posterior median is used instead.

Conditional on the sampled statistical model parameters, a prediction is sampled from the predictive distribution of $y_i^c(x, t)$. The sampling of parameters is repeated many times to obtain both $\mu_i^c(x, t)$ and $s_i^c(x, t)$ for the $i$-th computer model. Further details on prediction can be found in Chapter 3 and 5 for applications with and without obvious ranking, respectively.

The correlation term $\alpha_1(x, i)$ must also be evaluated as well. The Monte Carlo method
(Gelfand and Smith, 1990; Robert and Casella, 2005) is used to approximate this term such that

\[
\text{corr} \left[ y_f(x), y^c_i(x) \right] = \frac{E \left[ \left( y_f(x) - \mu_f(x) \right) \left( y^c_i(x) - \mu^c_i(x) \right) \right]}{s(x)s^c_i(x)}
\]  

(6.5)

\[
= \frac{1}{n} \sum_{j=1}^{n} \left( y_f(x)_j - \mu_f(x) \right) \left( y^c_i(x)_j - \mu^c_i(x) \right)
\]

where \( y^c_i(x)_j \) and \( y_f(x)_j \) are realizations from \( Y^c_i(x) \) and \( Y_f(x) \), respectively. In the examples we have explored, we found that \( n = 500 \) works reasonably well.

To begin using the EI in (6.2) and the criterion in (6.3), the response surface are estimated as outlined in Chapter 3 and 5. Similar to previous chapters, the design variables and calibration parameters are scaled to the unit interval. The computer models do not have to be ranked. However, suppose that there are two computer simulators. In the event that the computer models can be ranked, the first computer simulator represents the lower fidelity model while the higher fidelity computer code is referred to as the second computer simulator.

Given an initial design and the estimated response surfaces obtained through the methodologies proposed in Chapter 3 and 5, the proposed algorithm proceeds to sequentially add new simulations in six steps:

**Step 1.** Locate a potential region of where the global minimum is.

The main objective of this step is to explore the input space to identify potential regions of where \( x_{\text{min}} \) is. Denote \( x^f \) as the location of the smallest known field measurement, i.e. \( x^f = \arg\min_{x \in \{x_1, \ldots, x_n\}} y_f(x) \). The EI function is evaluated at \( \tau_1 \) design points that are in the neighbourhood of \( x^f \). We choose \( \tau_1 = 100 \) and the design points are generated from \( N(x^f, 0.1^2) \). The design point that maximizes the EI function is denoted as \( x^* \).

**Step 2.** Identify \( xVec^+ \) that maximizes (6.2).
The neighbourhood of \( x^* \) is investigated further. A set of 100 realizations from \( N(x^*, 0.05^2) \) is used in this step. The point with the greatest expected improvement is denoted as \( x^+ \).

**Step 3.** Select the most suitable computer model.

Conditional on the median of the posterior distribution of the calibration parameters obtained from the MCMC described in Chapter 3 and 5, depending on the type of applications, the criterion \( C(i|x^+) \) is evaluated for \( i = 1 \) and \( 2 \). The computer simulator that maximizes the criterion is denoted as \( i^+ \).

**Step 4.** Evaluate \( y_{i^+}^c (x^+, t) \).

**Step 5.** Estimate the response surfaces with the new data set.

The data is updated by including \( (x^+, y_{i^+}^c (x^+, t)) \) into the original data set. The response model is re-estimated and the posterior distributions of the calibration and statistical model parameters are obtained through MCMC.

**Step 6.** Iterate between steps 1 and 5 until the stopping criterion has been achieved.

The starting point of the search is changed by replacing \( x^f \) with \( x^+ \). Steps 1 to 5 are repeated until (6.3) in the previous and current iteration is less than \( \varepsilon \) three times. In our examples, we use \( \varepsilon = 0.05 \).

There is no guarantee that this approach will bring us to the exact location of global minimum of the physical system. Based on the added \( x^+ \) proposed by the methodology, a neighbourhood of \( x \) is formed to estimate the location of the minimum.

### 6.3 Examples

In this section, a few examples with multi-fidelity computer simulators will be used to illustrate the new algorithm. An example with computer simulators of similar fidelities will be discussed at the end of this section as well. Initially, the inputs for the simulated systems
CHAPTER 6. SEQUENTIAL KRIGING OPTIMIZATION

are sampled independently through the Latin hypercube sampling (Mackay et al., 1979). Then, the new algorithm will be employed to search for $x_{min}$ in the physical system.

### 6.3.1 A simple example

We begin with the example in Sasena et al. (2002), with some slight alterations. That is, the setting has been modified to accommodate two simulators and field experiments. In addition, we refashion the computer models to include a common calibration parameter and calibration parameters that exist in each simulator, respectively. The input settings are chosen from the interval $[0,10]$ whereas the calibration parameters comes from the unit interval.

The low fidelity model is defined as:

$$y^c_1(x,t_f,t_1) = \eta^c_1(x,t_f,t_1) = -\sin(x) - \exp\left\{\frac{x}{250t_f}\right\} + 50t_1.$$  \hfill (6.6)

Similar to Chapter 3, the high fidelity model is a linear combination of the low fidelity response model and a discrepancy term:

$$y^c_2(x,t_f,t_2) = \eta^c_1(x,t_f,\theta_1) + t_f t_2 + 0.03(x - 3) = \eta^c_1(x,t_f,\theta_1) + \delta_2(x,t_f,t_2).$$  \hfill (6.7)

Lastly, the observations are obtained by evaluating the mathematical model:

$$y^f(x) = \eta^c_1(x,\theta_f,\theta_1) + \delta_2(x,\theta_f,\theta_2) + 0.01x \log(x) + \epsilon = \eta^c_1(x,\theta_f,\theta_1) + \delta_2(x,\theta_f,\theta_2) + \delta_f(x) + \epsilon,$$  \hfill (6.8)

where $\epsilon \sim N(0, 0.5^2)$.

Following Loepky et al. (2009), we used a 30 run random Latin hypercube design (Mackay et al., 1979) for the low fidelity simulator. Only 10 runs were generated from (6.7) because the high fidelity model is considered to be more computationally expensive than the low fidelity model. The inputs for the high fidelity computer model were selected using
a random Latin hypercube design as well. Since, in practice, only a handful of experimental runs are available, 3 observations are obtained from (6.8).

We further assume that the cost of evaluating the low fidelity simulator to be 1 unit, i.e. $C^c_1 = 1$, whereas exercising the high fidelity simulator is more expensive such that $C^c_2 = 3$. Running the actual experiment usually requires more resources and hence, the cost per evaluation of (6.8) is chosen to be $C^f = 5$. These numbers are clearly made up and will change based on the application.

In this example, the true value of the common calibration parameter is chosen to be $\theta_f = 0.4$, while the calibration parameter appearing only in the high and low fidelity models are chosen to be $\theta_2 = 0.75$ and $\theta_1 = 0.2$, respectively. Figure 6.1 displays the response surfaces for the two simulators and also the mean response surface for the field process. These functions have two local minima. A quick glance at the figure reveals that the global minimum of the high fidelity model occurs near the global minimum of the physical system. The global minimum of the low fidelity model appears to be further away, but near a local minimum of the physical system. This represents the framework we are working within insofar as the high fidelity model is assumed to be more like the true system than the low fidelity model.

The new algorithm is used to search for the region where the global minimum of the physical system happens. In this example, $x_{\text{min}} \approx 1.5$. When a new simulation run is added to the simulated system, the posterior distribution of the model parameters are sampled using the MCMC outlined in Chapter 3.1.2. The MCMC chain is initialized with $\theta_f = \theta_2 = \theta_1 = 0.5$ (i.e., the centre of the input space), $\lambda_m = 1$, $\lambda_2 = \lambda_f = \lambda_y = 20$ and all the correlation parameters, $\rho$ are chosen to be 0.1 as we assume that the simulator and discrepancies are dependent on all the inputs. To determine the width of the Uniform distribution of the Metropolis and Hastings updates using the Graves method (Graves, 2011), 1,000 MCMC steps are used. The MCMC generally converges within 1,000 runs and hence, we run the MCMC for 11,000 steps and discarded the first 3,000 steps from further analysis.
When the difference between the current and previous augmented EI is less than 0.05 three times, the algorithm is terminated. The 8 new inputs added to the simulated models are summarized in Table 6.1 and represented graphically in Figure 6.1.

Although there are two local minima, the selected inputs are between 1.3 and 2, suggesting that the global minimum of interest is within this interval. This is a good indication that the proposed criterion is able to identify the global minimum within a few new simulation runs. The inputs around the other local minimum were not considered in this example. This is mainly due to the fact that the global minimum of the high fidelity simulator is very close to $x_{min}$, encouraging the algorithm to choose inputs around $x_{min}$.

The algorithm chooses the appropriate computer simulator to add the new runs based on the cost per evaluation of the computer simulators and the correlation between the simulated and physical systems. In this example, the correlation between the high fidelity
computer model and the true model is similar to the correlation between the low fidelity and field response surfaces. Under this circumstance, the algorithm will instinctively select the most cost efficient computer simulator, as shown in this example where 5 out of the 8 new simulations are added to the low fidelity computer model.

| Run | New Inputs, \((x, t_f, t_1)\) or \((x, t_f, t_2)\) | Computer Simulator | EI | \(C(i^+|\mathbf{x})\) |
|-----|---------------------------------|-------------------|----|------------------|
| 1   | \((1.379, 0.548, 0.5509)\)     | H                 | 3.1422 | 0.0323          |
| 2   | \((1.499, 0.5057, 0.1281)\)   | L                 | 4.0329 | 0.2326          |
| 3   | \((1.541, 0.58, 0.1381)\)    | L                 | 3.76  | 0.1504          |
| 4   | \((1.925, 0.4661, 0.1387)\)  | L                 | 3.984 | 0.1295          |
| 5   | \((1.727, 0.5707, 0.1636)\) | L                 | 2.9349 | 0.059          |
| 6   | \((1.905, 0.5023, 0.1679)\) | L                 | 2.8741 | -0.0251        |
| 7   | \((1.385, 0.4827, 0.4999)\) | H                 | 2.9116 | 0.0082        |
| 8   | \((1.346, 0.5273, 0.4653)\) | H                 | 2.5431 | -0.0211        |

Table 6.1: A summary of the new inputs added to the systems at each iteration. The high fidelity simulator is denoted with ‘H’ while the low fidelity simulator is represented with ‘L’.

### 6.3.2 Global minima in different locations

In the previous example, both the local minima of the physical system occur around the local optima of the high and low fidelity computer models. This feature is useful as the information coming from the computer models will guide the search to one of the local minima. To challenge the performance of the new algorithm, the \(x_{\text{min}}\) is made further away from the global minimum of the simulated models. The inputs are chosen from the unit interval to simplify the illustration of the example.

The low fidelity computer simulator is a mathematical function that has input \(x\), a common calibration parameter, \(t_f\) and a calibration parameter exclusive to itself, \(t_1\):

\[
g_1^c(x, t_f, t_1) = \eta_1(x, t_f, t_1) = \eta(1, t_f, t_1) = (x - t_f - 0.1)^2 + 0.3 \sin(20t_1x).
\]

By adding a discrepancy term \(\delta(x, t_f, t_2) = -t_2x(3t_f - x)\) to the low-fidelity simulator,
outputs for the high fidelity computer models are generated from:

\[ y_2(x, t_f, t_2) = \eta_1(x, t_f, \theta_1) + \delta_2(x, t_f, t_2). \]  

(6.10)

Lastly, some observations are available through:

\[ y^f(x) = \eta_1(x, \theta_f, \theta_1) + \delta_2(x, \theta_f, \theta_2) + x \log(x + 0.3) + \epsilon, \]  

(6.11)

where the measurement errors, \( \epsilon \), come from a Normal distribution with mean zero and standard deviation 0.2.

In this example, the true value of the calibration parameters are chosen to be \( \theta_f = 0.25 \), \( \theta_2 = 0.4 \) and \( \theta_1 = 0.25 \). The actual response surfaces of the field process, high and low fidelity models are shown in Figure 6.2. It is not hard to notice that the location of the global minimum of the physical system is further away from the global minima of the computer models. In fact, the global minima of the three systems are different and thus, increasing the difficulty of the search. This is the framework of interest. We want to investigate the influence of the computer models on the proposed approach and also, the ability of the algorithm to locate \( x_{\text{min}} \) in this circumstance.

Similar to the previous example, the cost-per-evaluation of the field experiment is 5 units and the cost to evaluate the high and low fidelity computer simulator is 3 units and 1 unit, respectively. The inputs for the three observations are obtained through Latin hypercube sampling (Mackay et al., 1979). The same method is used to sample 10 and 30 design points for the high and low fidelity models. The new algorithm is applied to identify the region where the global minimum of the physical system is and the MCMC described in Chapter 3.1.2 is used to sample the posterior distributions of the calibration and statistical model parameters after a new run is added to the simulated system.

The search ended after 4 iterations and the new inputs are summarized in Table 6.2. The new inputs appears to be close to the global minimum (see Figure 6.2). It may appear hasty to draw conclusion based on 4 new simulation runs. Hence, we assume that there unlimited resources and remove the default stopping criterion to let the algorithm add another 16 new
Figure 6.2: Response surfaces of the field process (blue), high fidelity (red) and low fidelity (green) model. The observations are in blue circles. The red and green ×’s are the design variables of the initial designs for the high and low fidelity simulators. The red and green circles at the bottom of the figure indicate the new $x$ added to the high and low fidelity simulated system, respectively, under the default stopping criterion.

inputs to the simulated systems. The inputs of the subsequent simulations are not far away from the inputs in the first 4 runs.

A further investigation is carried out to compare the effect of the new simulation runs, in particular the additional 16 runs, in identifying the global minimum of the physical process. The two models of comparison here are: (i) M4 – original model with the addition of the first 4 simulations; and (ii) M20 – original model with all 20 simulation runs obtained in Table 6.2. The MCMC described in Chapter 3.1.2 is used to sample the posterior distributions of the calibration and statistical model parameters of M4 and M20. An independent design of size 25 is obtained through Latin hypercube sampling (Mackay et al., 1979) and both models are used to make prediction of the field process at the 25 points. The predictions are shown in Figure 6.3.

Notice that the predictions at smaller $x < 0.65$ are terrible. This is not a major concern here because our goal is to identify the global minimum of the field process. The predictions
of M4 at $x \geq 0.65$ turn out to be better than the predictions of M20. Although both models show their global minima around the true global minimum, the minimum of model M4 appears to be closer to the truth. The additional 16 computer runs cause M20 to rely heavily on the computer simulators such that it ends up modelling the high fidelity computer code.

Oftentimes, the features of the field process are unknown and the computer models are unable to capture reality in a perfect manner. When it is not feasible to exercise the computer simulators for a long time due to reasons such as cost and time, the proposed algorithm with the default stopping criterion can provide useful insight to identifying the feature of the physical system.

| Run | New Inputs, ($x, t_f, t_1$) or ($x, t_f, t_2$) | Computer Simulator | EI | $C(i^+|x)$ |
|-----|---------------------------------------------|-------------------|----|------------|
| 1   | (0.7208, 0.2557, 0.4669)                    | H                 | 0.3211 | 0.001     |
| 2   | (0.6858, 0.309, 0.2649)                     | L                 | 0.281  | 0.0098    |
| 3   | (0.7556, 0.2733, 0.439)                    | H                 | 0.2895 | 0.0031    |
| 4   | (0.7415, 0.2503, 0.2347)                    | L                 | 0.2702 | 0.0127    |
| 5   | (0.729, 0.2581, 0.2263)                     | L                 | 0.2873 | 0.0098    |
| 6   | (0.7284, 0.2495, 0.2687)                    | L                 | 0.2531 | 0.0076    |
| 7   | (0.696, 0.2603, 0.4064)                     | H                 | 0.2953 | -0.0047   |
| 8   | (0.7215, 0.2599, 0.273)                     | L                 | 0.2961 | 0.0067    |
| 9   | (0.7365, 0.2429, 0.2751)                    | L                 | 0.2403 | 0.0069    |
| 10  | (0.7415, 0.2258, 0.459)                     | H                 | 0.2718 | 0.0032    |
| 11  | (0.6973, 0.2628, 0.5064)                    | H                 | 0.2813 | 0.0015    |
| 12  | (0.6884, 0.2737, 0.2309)                    | L                 | 0.2419 | -0.0016   |
| 13  | (0.7386, 0.2708, 0.2327)                    | L                 | 0.2264 | 0.0028    |
| 14  | (0.7357, 0.3041, 0.2451)                    | L                 | 0.2366 | 0.0044    |
| 15  | (0.7685, 0.2406, 0.2505)                    | L                 | 0.225  | 0.009     |
| 16  | (0.6691, 0.2818, 0.4829)                    | H                 | 0.2454 | 0.0012    |
| 17  | (0.7651, 0.2481, 0.4744)                    | H                 | 0.2249 | -0.0006   |
| 18  | (0.7296, 0.245, 0.2721)                     | L                 | 0.2257 | 0.0088    |
| 19  | (0.7659, 0.2385, 0.258)                     | L                 | 0.2184 | 0.0052    |
| 20  | (0.8194, 0.25, 0.2614)                      | L                 | 0.2599 | 0.0107    |

Table 6.2: A summary of the new inputs added to the systems at each stage. The high fidelity simulator is denoted with ‘H’ while the low fidelity simulator is represented with ‘L’. The horizontal line denotes the cut-off determined by the default stopping criterion.
6.3.3 Global minimum in an unsampled region

The observations in the previous examples discussed thus far are sampled uniformly in the input space. This, however, is not plausible in some applications when collecting experimental results from certain parts of the input space is difficult or impossible. There is a possibility that the input space explored by the experimenter does not contain the global minimum of the field process. Hence, we will investigate the ability of the algorithm to utilize information from the computer models to locate the global minimum of the physical system in such circumstances.

We continue to use the mathematical model in (6.10) to generate outputs for the low fidelity simulated system. The discrepancy term \( \delta_2(x, t_f, t_2) = -t_f t_2 x \log(x) \) is added to the low-fidelity simulator when generating outputs for the high fidelity computer models, i.e

\[
y^c_2(x, t_f, t_2) = \eta_1(x, t_f, \theta_1) + \delta_2(x, t_f, t_2).
\]  

The observations, containing measurement errors \( \epsilon \sim \text{iid} \ N(0, 0.2^2) \), come from a combination
of the low fidelity simulator and two discrepancy terms:

\[ y^f(x) = \eta_1(x, \theta_f, \theta_1) + \delta_2(x, \theta_f, \theta_2) + 0.15x(0.5 - x) + \epsilon. \] (6.13)

The inputs for the 10 high fidelity and 30 low fidelity simulation runs are selected using Latin hypercube sampling (Mackay et al., 1979) as mentioned in the previous examples. For the purpose of simplicity, the inputs are chosen from the unit interval. To create the situation of interest, the inputs of the three observations are chosen to be \( x = 0.1, 0.2 \) and \( 0.3 \) and experts are assumed to have no knowledge of the physical system when \( x \in (0.3, 1] \).

A quick glance in Figure 6.4 shows that the global minimum is around 0.78, which is not in the interval [0, 0.3].

![Figure 6.4: Response surfaces of the field process (blue), high fidelity (red) and low fidelity (green) model. The observations are in blue circles, whereas the red and green x’s denote the design variables of the initial designs for the high and low fidelity simulators. The red and green circles at the bottom of the figure indicate the new x added to the high and low fidelity simulated system, respectively, under the default stopping criterion.](image)

The cost-per-evaluation of the simulated and physical systems remains 5 units for field experiment, 3 units for the high fidelity simulator and 1 unit for the low fidelity simulator. To identify \( x_{min} \), we applied the proposed algorithm as described earlier and included the MCMC mentioned in Chapter 3.1.2 as required.
The search, in this example, also terminated after 4 iterations. Based on our experience from the last example, no additional runs will be added. Looking at Table 6.2, we notice that the algorithm started off by choosing the input closer to the interval \([0, 0.3]\) before moving towards the location of the global minimum. The result is promising. It shows that the computer simulators provided sufficient information for the algorithm to explore the domain unknown to the experimenters. In this example, we are able to verify the result with the response surface in Figure 6.4. However, it is difficult to visualize the response surface of the physical system in practice. Experimenters will need to collect field measurements in the neighbourhood of \(x_{\text{min}}\) to confirm their findings.

| Run | New Inputs, \((x, t_f, t_1)\) or \((x, t_f, t_2)\) | Computer Simulator | EI   | \(C(i^+ | \x)\) |
|-----|---------------------------------|-----------------|------|--------------|
| 1   | \((0.5755, 0.2697, 0.2109)\)    | L               | 0.3211| 0.2133       |
| 2   | \((0.7493, 0.2673, 0.5214)\)    | H               | 0.281 | 0.4857       |
| 3   | \((0.7748, 0.2722, 0.5796)\)    | H               | 0.2895| 0.3913       |
| 4   | \((0.7744, 0.2773, 0.5161)\)    | H               | 0.2702| 0.3054       |

Table 6.3: The new inputs added to the systems at each iteration. The high fidelity simulator is denoted with ‘H’ while the low fidelity simulator is represented with ‘L’. The horizontal line denotes the cut-off determined by the default stopping criterion.

### 6.3.4 A bad low fidelity model

So far, the low fidelity computer models in Example 6.3.1, 6.3.2 and 6.3.3, although not as good as the high fidelity model, resemble the true model quite well. It is not hard to imagine a situation where the low fidelity simulator is uninformative and the discrepancy between the two computer simulators is large. The course of action for the algorithm when this situation arises is of interest, and this example aims to answer our curiosity. To simplify the illustration of this example, the inputs \(x\) and \(t\) are chosen from the unit interval \([0, 1]\).

To create an extreme case, the low-fidelity computer model is chosen to be a horizontal
\begin{equation}
yc_1(x, t_f, t_1) = \eta_1(x, t_f, t_1) = t_f + t_1.
\end{equation}

The low fidelity model is only a function of the calibration parameters and will not be influenced by the input $x$ at all. The high fidelity simulator, which has a closer resemblance of the actual process, will have a large discrepancy with the low fidelity model. Our choice of discrepancy term is $\delta_2(x, t_f, t_2) = t_2 \sin(20xt_f)$, and again, the simulated responses from the high fidelity simulator is a linear combination of $\eta_1(\cdot)$ and $\delta_2(\cdot)$:

\begin{equation}
yc_2(x, t_f, t_2) = \eta_1(x, t_f, \theta_1) + \delta_2(x, t_f, t_2).
\end{equation}

Lastly, a handful of observations, with measurement errors $\epsilon \sim iid N(0, 0.5^2)$, are obtained through:

\begin{equation}
y_l(x) = \eta_1(x, \theta_f, \theta_1) + \delta_2(x, \theta_f, \theta_2) - (x - 0.33)0.2x + \epsilon.
\end{equation}

The true calibration parameters in this example are $\theta_f = 0.25, \theta_2 = 0.3$ and $\theta_1 = 0.1$. The response surfaces of the field experiment, high and low fidelity computer simulators are shown in Figure 6.5. The low fidelity computer model (in green) looks very different from both the high fidelity computer code (in red) and the true surface (in blue). This is the framework of interest and the algorithm is challenged to search for the location of the global minimum of the physical process, i.e. $x_{\text{min}} \approx 0.9$.

As in the previous examples, 30 outputs are simulated from (6.15) and 10 outputs are obtained through (6.15). Lastly, three observations are generated from (6.16). The observations are sampled uniformly from the unit interval while the inputs for both the computer models are sampled through independent Latin hypercube sampling (Mackay et al., 1979).

The cost-per-evaluation of the three systems remain the same as in the previous examples, and the proposed algorithm is applied to this example. After 4 iterations, the algorithm
Figure 6.5: The blue solid line is the function of the field experiment. The surfaces of the high and low fidelity simulators are shown in red and green dotted lines. The observations are in blue circles, whereas the red and green $\times$’s are the design variables of the initial designs for the high and low fidelity simulators. The red and green circles at the bottom of the figure indicate the new $x$ added to the high and low fidelity simulated system, respectively.

terminated and suggested that $x_{\text{min}}$ lies between 0.84 and 0.91. This is a reasonable estimate of the global minimum of the field process.

A closer inspection of Table 6.4 reveals that the algorithm started off by adding one new low fidelity simulation run to the system, but the next three newly added simulated outputs came from the high fidelity computer code. This shows that the algorithm recognized the information coming from the low fidelity computer model is unreliable and chose to use the high fidelity simulated simulator.

| Run | New Inputs, $(x,t)$ | Computer Simulator | EI  | $C(t^+|x)$ |
|-----|---------------------|--------------------|-----|------------|
| 1   | $(0.8441, 0.3907, 0.3549)$ | L                  | 0.0255 | 0.0574     |
| 2   | $(0.8551, 0.3526, 0.4089)$ | H                  | 0.0165 | 0.0152     |
| 3   | $(0.8843, 0.3761, 0.426)$  | H                  | 0.0011 | 0.0009     |
| 4   | $(0.9121, 0.3701, 0.4079)$ | H                  | 0.0100 | 0.0081     |

Table 6.4: Proposed new runs added to the simulators.
The result shows that the proposed algorithm is capable of leading the search to the region where the global minimum of the true process is. We also find that this new algorithm identified the low fidelity computer model as questionable and excluded it from adding further simulation run. This concludes that the new algorithm works reasonably well for applications with multi-fidelity computer simulators.

6.3.5 Computer simulators of similar fidelities

The setting of this example is the same as that discussed in Chapter 5. The mathematical model used in this example is also an adaptation of an example from Sasena et al. (2002). Our modification creates two computer simulators that depend on an input $x$ and a calibration parameter $\theta$.

The observations are considered to be noisy version of the perfect simulator, $\zeta^f(\cdot)$. The measurement errors, $\epsilon$, are normally distributed with mean zero and standard deviation 0.5. In short, the experimental values come from:

$$
\begin{align*}
y^f(x) &= \zeta^f(x, \theta) \\
&= -\sin(x) - \exp\left(\frac{x}{250t}\right) + 10 + \epsilon, \\
&= (6.17)
\end{align*}
$$

Both computer models $\eta^1_c(\cdot)$ and $\eta^2_c(\cdot)$ are linear combination of the perfect simulator and two independent bias terms $\Delta_1(\cdot)$ and $\Delta_2(\cdot)$. The outputs from the first and second computer simulators are simulated from

$$
\begin{align*}
y^c_1(x, t) &= \zeta^f(x, t) + \Delta_1(x) \\
&= \zeta^f(x, t) + 0.3 + 0.03(x - 3), \\
&= (6.18)
\end{align*}
$$

and

$$
\begin{align*}
y^c_2(x, t) &= \zeta^f(x, t) + \Delta_2(x) \\
&= \zeta^f(x, t) - \Delta_1(x). \\
&= (6.19)
\end{align*}
$$
The true value of the calibration parameter in this example is $\theta = 0.4$ and the response surfaces of the physical and simulated systems are shown in Figure 6.6. A quick glance at the figure shows that both the computer codes have about the same fidelity. The actual system and the second computer simulator have their global minimum around 8, whereas the first computer simulator has its global minimum further away from the other two systems.

In this example, we generated 20 outputs for each of the first and second computer models and obtained 3 observations from (6.17). To create the framework of interest, we chose the cost-per-evaluation of both computer models to be the same, i.e. $C_1^c = C_2^c = 1$. The cost to perform an experimental run is chosen to be $C^f = 5$, making it more expensive than to exercise the computer codes.

The new algorithm is used to search for the region where the global minimum of the
physical system happens. When a new simulation run is added to the simulated system, the posterior distributions of the model parameters are sampled using the MCMC outlined in Chapter 5. The MCMC chain is initialized with $\theta_f = \theta_2 = \theta_1 = 0.5$ (i.e., the centre of the input space) and $\lambda_\zeta = 1$, $\lambda_1 = \lambda_2 = \lambda_y = 20$. The correlation parameters, $\rho$ are chosen to be 0.1 as we assume that the simulator and discrepancies are dependent on all the inputs. To determine the width of the Uniform distribution of the Metropolis updates using the Graves method (Graves, 2011), 1,000 MCMC steps are performed. The MCMC generally converges within 1,000 runs and hence, we run the MCMC for 11,000 steps and discarded the first 2,000 steps from further analysis.

We chose to terminate the algorithm when the difference between the current and previous augmented EI is less than 0.05 three times. The 5 new inputs added to the simulated models are summarized in Table 6.5 and represented graphically in Figure 6.6.

| Run | New Inputs, $(x,t)$ | Computer Simulator | EI  | $C(i^+ | x)$ |
|-----|-------------------|-------------------|-----|-------------|
| 1   | (0.8302, 0.4736)  | 2                 | 0.4401 | 0.1095     |
| 2   | (0.8071, 0.5553)  | 1                 | 0.3272 | 0.0818     |
| 3   | (0.8024, 0.2113)  | 1                 | 0.3566 | 0.0828     |
| 4   | (0.7946, 0.4799)  | 2                 | 0.3351 | 0.0837     |
| 5   | (0.7840, 0.4948)  | 1                 | 0.2976 | 0.0828     |

Table 6.5: Proposed new runs added to the simulators.

The new simulations added to the systems have inputs between 7.8470 and 8.302. This interval contains the location of the global minimum of the field process. It does not go unnoticed that the new simulations come from both computer models. This is mainly due to the similarity in fidelity and cost associated to the respective simulators.

The result from this example is a good indication that the proposed methodology can be used for applications with multiple computer simulators of similar fidelity. In general, this new algorithm can be applied to applications with outputs coming from different computer models. The fidelity and cost of each simulator will affect the decision of the algorithm on
the choice of simulator to generate new outputs from.

\section{Discussion}

In this section, some extensions of the proposed approach are discussed while some avenues for future work are identified. So far, the focus has been on the setting where there are only two simulators. The new methodology, however, can easily be extended to model applications that involve more than two simulators.

The proposed search algorithm is a simple random walk that is intuitive and easy to implement. It has worked well in the examples that we have explored thus far. However, more complicated response surfaces are expected to call for more sophisticated optimization algorithms such as Nelder-Mead simplex method (Nelder and Mead, 1965) and the branch-and-bound algorithm (Robert and Casella, 1995). Both the EI function in (6.2) and criterion in (6.3) can be maximized using readily available optimizers in common software packages such as Matlab and R as well.

Similar to most optimization problems, the identification of the global extremum depends on the starting values. To identify the starting points for the optimization routines, Ranjan et al. (2008) proposed a genetic algorithm (e.g. Holland (1975); Mandal et al. (2006)) that has shown to be computationally fast.
Chapter 7

Conclusion

In this thesis, methodologies were developed for prediction, model calibration and optimization using outputs from multiple computer models. A new methodology, which combines outputs from multi-fidelity simulators and field observations, was proposed in Chapter 3. The approach successfully uses a Bayesian hierarchical model to make predictions of the physical system with associated measurements of uncertainty (e.g., posterior variance or prediction intervals). Different Gaussian processes are used to model the various response surfaces. The real example that motivated this work used two simulators of the process, but methodology can be easily extended to cases with more than two simulators.

The new Bayesian approach presented in Chapter 4 moves the discrepancy term to the level of the computer simulator. The model puts together information from the physical process and a single computer simulator to make prediction and estimate the calibration parameters. In the examples we have considered, the new model performed as well as the methodology proposed by Kennedy and O’Hagan (2001). This new methodology is further extended to model outputs from multiple computer simulators that have similar or unclear fidelities. The response surfaces of the physical and simulated systems are modelled with independent Gaussian processes. Reliable predictions has been obtained through this model in the examples.
We also proposed a sequential design strategy to identify features of the response surface of the physical process in Chapter 6. The inputs for the new simulation runs are selected using the expected improvement function while a new criterion is used to choose the most cost efficient computer simulator. The new algorithm has been shown to work reasonably well in a few simple examples that contain multiple computer simulators of similar or different fidelities.

The new methodologies in this thesis bring forth many avenues for future research. Modelling the response surfaces in Chapter 3 and 5 with non-stationary GPs is a natural extension while the sampling of the posterior distributions of the statistical and calibration parameters can be replaced with more efficient MCMC procedures. The same can be said about the search algorithm for the global extremum of the field process when there is more than one computer simulator.


