Modelling of Dynamic Computer Experiments with Both Qualitative and Quantitative Variables

by

Jingyi Liang

A thesis submitted to the

Department of Mathematics & Statistics

in conformity with the requirements

for the degree Masters of Science

Queen's University

Kingston, Ontario, Canada

January 2018

Abstract

Computer experiments are utilized as a popular tool of studying the relationship between responses and the factors that affect them. They are widely used in scientific researches and applications. Dynamic computer experiments refer to computer experiments with time-series responses. Different dynamic computer models have been proposed to emulate the relationship between the time-series responses and the corresponding quantitative factors. Qualitative factors are also widely used and show important effects in many scientific problems. Different models have been proposed for one-dimensional outputs and both quantitative and qualitative factors.

In this thesis, we have reviewed some popular dynamic computer models with only quantitative factors. We make empirical comparisons of the prediction accuracy of three existing dynamic computer models with only quantitative factors. Among the three models, the dynamic linear Gaussian process model and the singular value decomposition based Gaussian process model outperform. Based on these two models, we incorporate the qualitative factors utilizing the additive correlation structure and propose two new modelling methods for dynamic computer experiments with both quantitative and qualitative factors. We also explored the different choices of the prior distributions for the unknown parameters in these two models. The prediction accuracy of the two proposed modelling approaches is tested by a limited simulation

study.

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr. Devon Lin and Dr. Glen Takahara. Dr. Lin has been guiding me throughout my research process. I truly appreciate her advice and support during the past two years. Her guidance has kept me focused and made my writing process less painful. Thanks to her weekly meetings, I am able to keep my research on track and figure out my questions as soon as possible. I also appreciate Dr. Takahara's advice and thorough review of this thesis.

I am also lucky to have great friends shown their support to me whenever I need. In particular, thanks to my friend Joanna, for being my greatest support. Thanks to my friend Emily, who studied with me together throughout my graduate period. Thanks to my friends May, Christina, Ru, Zoey, Linkin and Bing who helped me make my life at Queen's University a better one. Also, thanks to my friends from my undergraduate period, Bowen, Yingchun, Anan, Yufei, Yuxiang, Rui and Menghua for all the funny conversations and encouragements.

I am thankful for the entire Department of Mathematics and Statistics of Queen's University. I would like to thank all professors from whom I have learned a lot. I would also like to thank Jennifer Read for her assistance over the past two years.

Last, and most importantly, I would like to thank my parents for being my biggest

support, morally and financially. I know you have sacrificed a lot so that I can be who I am today. I am really appreciative.

Contents

List of Figures

Chapter 1

Introduction

Computer experiments use computer simulators to construct the relationship between inputs and the corresponding responses on computers [23]. A computer simulator is usually a piece of computer code. Computer experiments are increasingly popular nowadays and exert a significant effect on a variety of scientific researches and applications. In many cases, computer experiments are integrant when physical experiments are too time-consuming, too expensive or just impossible like the climate modelling [24]. In a computer experiment, the function that produces the corresponding responses of inputs is usually assumed to be unknown, although the outputs can be obtained with some given inputs.

Dynamic computer experiments refer to computer experiments with time series outputs. When a simulator is computationally demanding to produce few evaluations, we need the emulator, which enables the approximation of the original computer simulator. An emulator is a statistical model approximating the relationship between inputs and outputs. Emulators maintain the original key features and behaviour but are always computationally attractive when the required time of simulator evaluations is long. Statistical emulators of computer simulators have proven to be useful in a variety of applications, for example, the geological and climate study [23].

One key characteristic of computer experiments is that they are deterministic, and that is the main difference between computer experiments and physical experiments. We call a simulator deterministic, if given a particular input, the simulator will always produce the identical output. This characteristic means that there is no random error term in a computer model, thus, the replication in the physical experiment is not applicable in computer experiments.

The input factors in computer experiments can be quantitative or qualitative. Different simulators designed for only quantitative factors were introduced in the literatures ([23], [24], [4], [9], [11], [17]). Qualitative factors are also widely used and show important effects in many scientific problems and applications [7]. Frequently computer experiments involve both quantitative and qualitative factors. Computer experiments with both quantitative and qualitative factors were studied in recent research ([26], [21]).

One main goal in computer experiments is to predict the unknown responses at some untried inputs. To achieve the goal, we need to build a statistical model that approximates relationship between inputs and outputs via some known inputs and their corresponding responses. A fundamental question in computer experiments is how to select the settings of input variables to run the computer code, so that the obtained data allow a cheaper surrogate model to be best constructed. The settings of these input variables give a set of design points that must be selected with great care since running the sophisticated computer code normally requires several hours, and in some cases several days, to produce one single output. Space-filling designs are commonly used by scientists as they provide a good coverage of the design space by spreading the designs points evenly in that space. One commonly used class of space-filling designs is Latin hypercube design, introduced by [18]. When a Latin hypercube design $\boldsymbol{D}_{n\times p}$ projects to any of the p dimensions of the input, the n projection points are located in all n equally spaced intervals and each interval contains only one point. For qualitative factors, we use the fractional factorial design [16]. A fractional factorial design consists of a carefully chosen subset of a full factorial design. A full factorial design means that all possible combinations of the qualitative factors are included in the design. A fractional factorial design can carry out the information of important features in the data. In the simulation study, we use the method introduced by [15] to generate the qualitative factors.

In recent years, there are many developments in dynamic computer experiments. For dynamic models with quantitative factors only, [11] proposed singular value decomposition (SVD)-based Gaussian process (GP) models with high-dimensional output. [4] proposed three methods to build emulators for dynamic computer models: multivariate output emulators, many single-output (MS) emulators, and time-input (TI) emulator. Multivariate output emulator was built based on GP with multivariate outputs. MS emulator was to find the relationship between the design matrix and subset of output matrix, i.e. the outputs at the same time. For each subset of outputs at the same time, a univariate output GP emulator is built. TI emulator was to include time factor as an input and build a single output emulator. [17] built the model combining the dynamic linear model and GP by extending the time-varying autoregressive model and utilizing GPs to describe the relationship between different time series.

For computer experiments with both quantitative and qualitative factors, [19] proposed a framework to incorporate qualitative factors with special structures into GP model. [21] improved GP model to accommodate general structures of qualitative factors by building an unrestrictive correlation structure of quantitative and qualitative factors. [21] assumed that the correlation functions of both types of factors are multiplicative. [26] constructed the model with additive GPs and combined the correlation function between qualitative factors using additive structure. [26] modelled the correlation for each qualitative factor and all quantitative factors with an independent GP, and assumed the correlation functions of different qualitative factors are additive. To overcome the difficulty of estimating the correlation matrix, [26] used the hypersphere decomposition method to build the correlation matrix proposed by [25] to ensure the positive-definiteness of the correlation structure.

In the thesis, the goal is to build emulators for dynamic computer experiments with both quantitative and qualitative factors. The proposed approaches are to incorporate the additive correlation structure of qualitative factors into simulators with only quantitative factors. We propose two new modelling methods by extending two existing models with quantitative factors, the dynamic linear models with GPs [17] and SVD based GP model [11], to incorporate qualitative factors.

In Chapter 2, we review the background and recent researches on the dynamic computer experiments. Chapter 3 makes a comparison of MS, TI model in [4], dynamic linear models with GPs in [17] and SVD based GP model in [11] with only quantitative factors. Chapter 4 presents the proposed two new models with both quantitative and qualitative factors. Chapter 4.1 introduces dynamic linear model with additive GP, and Chapter 4.2 introduces the SVD based additive GP model.

Chapter 4.3 is the simulation study of two proposed models. Concluding remarks are given in Chapter 5.

Chapter 2

Literature Review

This chapter reviews the existing models on dynamic computer experiments with only quantitative factors and with both quantitative and qualitative factors based on GP models [24].

In Section 2.1, GP with univariate output is reviewed. In Section 2.2, different models with multivariate output based on GP with quantitative factors are reviewed. Section 2.3 reviews the additive GP model with quantitative and qualitative factors.

2.1 Models with Quantitative Factors and Univariate Output

In this section, we review the GP-based model with univariate output. Suppose $\chi \subset \mathbb{R}^p$ is a p-dimensional input space. For $\boldsymbol{x} \in \chi$, $y(\boldsymbol{x})$ is a Gaussian random function [24] if $(y(\boldsymbol{x}_1), ..., y(\boldsymbol{x}_n))$ has a multivariate normal distribution for any $\boldsymbol{x}_1, ...\boldsymbol{x}_n \in \chi$ and $n > 1$. A Gaussian random function is determined by the mean function and the covariance function

$$\begin{aligned}\mu(\boldsymbol{x}) &= E\{y(\boldsymbol{x})\} \\ \mathrm{Cov}\{y(\boldsymbol{x}_1), y(\boldsymbol{x}_2)\} &= \sigma^2 R(\boldsymbol{x}_1 - \boldsymbol{x}_2)\end{aligned} \tag{2.1}$$

in which σ^2 is the variance and $R(\cdot)$ is the correlation function.

The GP model generally treats the outputs as a realization of a random function:

$$y(\boldsymbol{x}_i) = \sum_{j=1}^{m} f_j(\boldsymbol{x}_i)\beta_j + \varepsilon(\boldsymbol{x}_i) = \boldsymbol{f}'(\boldsymbol{x}_i)\boldsymbol{\beta} + \varepsilon(\boldsymbol{x}_i) \tag{2.2}$$

for $1 \leq i \leq n$, where the $\{f_j(\cdot)\}, 1 \leq j \leq m$ are known regression functions, $\boldsymbol{\beta} = (\beta_1, ..., \beta_m)'$ is an $m \times 1$ vector of regression coefficients, and $\varepsilon(\cdot)$ is a zero mean GP with the covariance function of specified by (2.1) for some correlation function $R(\cdot)$ and the variance σ^2. The correlation function can naturally be denoted as $R(\boldsymbol{x}_i - \boldsymbol{x}_j) = R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is a vector of correlation parameters. One widely used form of the correlation function is power-exponential correlation function, where the correlation parameters are represented by $\boldsymbol{\phi} = (c_1, ..., c_p, \alpha_1, ..., \alpha_p)$

$$R(\boldsymbol{x}_i - \boldsymbol{x}_j) = \exp(-\sum_{k=1}^{p} c_k |x_{ik} - x_{jk}|^{\alpha_k}) \tag{2.3}$$

for any two inputs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $1 \leq i, j \leq n$, with the range parameter c_k's and the roughness parameters α_k's. Among all choices of α_k's and c_k's, $\alpha_i = a$ for all i and $a = 2$ is a popular choice, so that the process is infinitely differentiable at the origin [24].

Consider a simulator that takes n p-dimensional inputs

$$\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)' = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}. \tag{2.4}$$

We want to predict the response at a new input point $\boldsymbol{x}_0$, the joint distribution of $y_0 = y(\boldsymbol{x}_0)$ and $\boldsymbol{y} = (y(\boldsymbol{x}_1), ..., y(\boldsymbol{x}_n))'$ is the multi-variate normal distribution

$$\begin{pmatrix} y_0 \\ \boldsymbol{y} \end{pmatrix} \sim N_{n+1}\left[\begin{pmatrix} \boldsymbol{f}_0' \\ \boldsymbol{F} \end{pmatrix}\boldsymbol{\beta}, \sigma^2 \begin{pmatrix} 1 & \boldsymbol{r}_0' \\ \boldsymbol{r}_0 & \boldsymbol{R} \end{pmatrix}\right] \tag{2.5}$$

where $\boldsymbol{f}_0 = \boldsymbol{f}(\boldsymbol{x}_0)$ is the $m \times 1$ vector of regressors at $\boldsymbol{x}_0$, $\boldsymbol{F}$ is the $n \times m$ matrix of regressors $\boldsymbol{F} = (\boldsymbol{f}(\boldsymbol{x}_1), ..., \boldsymbol{f}(\boldsymbol{x}_n))'$, $\boldsymbol{\beta}$ is an $m \times 1$ vector of regression coefficients, $\boldsymbol{r}_0$ is an $n \times 1$ vector $\boldsymbol{r}_0 = (R(\boldsymbol{x}_0 - \boldsymbol{x}_1), ..., R(\boldsymbol{x}_0 - \boldsymbol{x}_n))'$, and $\boldsymbol{R}$ is an $n \times n$ matrix with (i, j)-th element $\boldsymbol{R}_{i,j} = R(\boldsymbol{x}_i - \boldsymbol{x}_j)$ defined by the correlation function. The conditional distribution of the new output is normal with mean

$$\hat{y_0} = E\{y_0 | \boldsymbol{y}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\phi}\} = \boldsymbol{f}_0'\boldsymbol{\beta} + \boldsymbol{r}_0'\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}) \tag{2.6}$$

and variance

$$\mathrm{Var}\{y_0 | \boldsymbol{y}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\phi}\} = \sigma^2(1 - \boldsymbol{r}_0'\boldsymbol{R}^{-1}\boldsymbol{r}_0).$$

In practice, $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\phi}$ are usually unknown and need to be estimated. This can be achieved using maximum likelihood estimation (MLE) [23] and Bayesian method [5]. Here we review the MLE method in details. Using the MLE method, we obtain the estimate of the parameters by maximizing the log-likelihood function

$$l(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\phi}|\boldsymbol{y}) = -\frac{n}{2}\log(2\pi\sigma^2) - \log|\boldsymbol{R}(\boldsymbol{\phi})| - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta})'\boldsymbol{R}(\boldsymbol{\phi})^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}).$$

The usual generalized least-squared estimate of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{y}. \tag{2.7}$$

Given $\hat{\boldsymbol{\beta}}$, the maximum likelihood estimators of σ^2 and $\boldsymbol{\phi}$ are

$$\hat{\boldsymbol{\phi}} = \text{argmax}\{-n\log\hat{\sigma}^2(\boldsymbol{\phi}) + \log(\mathbf{det}\,(\boldsymbol{R}))\} \tag{2.8}$$

and

$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})'\boldsymbol{R}(\hat{\boldsymbol{\phi}})^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}). \tag{2.9}$$

2.2 Models with Quantitative Factors and Multivariate Outputs

In this section, we review the models with multivariate outputs and only quantitative inputs. In Section 2.2.1 to Section 2.2.5, we review the multivariate GP model [4], SVD based model [11], MS model [4], TI model [4], and the dynamic linear GP model [17] respectively.

2.2.1 Model with Multivariate Outputs Based on Multivariate Gaussian Process

In this section, we review the multivariate output model introduced by [4]. We consider a simulator with p-dimensional input $\boldsymbol{x} \in \chi$ and multivariate T-dimensional output $\boldsymbol{y} = (y_1(\boldsymbol{x}), ..., y_T(\boldsymbol{x}))'$. This is called multivariate output model. A dynamic simulator is one special kind of multivariate output models, each of whose outputs is a T-dimensional time series.

Assume $\boldsymbol{X}$ is an $n \times p$ design matrix as (2.4) and the corresponding output matrix

$$\boldsymbol{Y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)' = \begin{pmatrix} y_1(\boldsymbol{x}_1) & \cdots & y_T(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ y_1(\boldsymbol{x}_n) & \cdots & y_T(\boldsymbol{x}_n) \end{pmatrix}_{n \times T}. \tag{2.10}$$

We also define $\boldsymbol{y}(\boldsymbol{x}) = (y_1(\boldsymbol{x}), ..., y_T(\boldsymbol{x}))$. Assume the simulator is a function $\boldsymbol{y}(\cdot) : \boldsymbol{X} \to \mathbb{R}^T$, we regard $\boldsymbol{y}(\cdot)$ as an unknown function

$$\boldsymbol{y}(\cdot)|\boldsymbol{B}, \boldsymbol{\Sigma} \sim GP(\boldsymbol{m}(\cdot), \boldsymbol{\Sigma})$$

where $\boldsymbol{m}(\cdot)$ is the mean function, $\boldsymbol{\Sigma}$ is the covariance matrix and $\boldsymbol{B}$ is a matrix of regression coefficients. That is to say, the distribution of $\boldsymbol{y}(\boldsymbol{x}_i)$ is matrix variate normal, with mean and variance

$$\begin{aligned} \boldsymbol{m}(\boldsymbol{x}_i) &= E[\boldsymbol{y}(\boldsymbol{x}_i)|\boldsymbol{B}, \boldsymbol{\Sigma}] = \boldsymbol{B}'\boldsymbol{h}(\boldsymbol{x}_i), 1 \leq i \leq n, \\ \text{Cov}\left[\boldsymbol{y}(\boldsymbol{x}_i), \boldsymbol{y}(\boldsymbol{x}_j)|\boldsymbol{B}, \boldsymbol{\Sigma}\right] &= \boldsymbol{\Sigma}_{i,j}, 1 \leq i, j \leq n, \end{aligned} \tag{2.11}$$

where $\boldsymbol{h}(\cdot)$ is the known regression function.

One way to describe the covariance matrix is to treat it as a separable form as a product of between-input covariance and between-output covariance [4]. That is,

$$\boldsymbol{\Sigma} = c(\cdot,\cdot)\boldsymbol{C}.$$

where $c(\cdot,\cdot)$ describes the between-input covariance depending on $(\boldsymbol{x}-\boldsymbol{x}'), \boldsymbol{x}, \boldsymbol{x}' \in \chi$, and $\boldsymbol{C}$ describes the between-output covariance. That is to say, the covariance in (2.11) can be written as

$$\boldsymbol{\Sigma}_{i,j} = \mathrm{Cov}\left[\boldsymbol{y}(\boldsymbol{x}_i), \boldsymbol{y}(\boldsymbol{x}_j)|\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\alpha}\right] = c(\boldsymbol{x}_i, \boldsymbol{x}_j)\boldsymbol{C}$$

conditional on hyper-parameters $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{\alpha}$, where $\boldsymbol{B} \in \mathbb{R}^{m\times T}$ is a matrix of regression coefficients; $\boldsymbol{C}$ represents the between-output covariance matrix; $diag(\boldsymbol{\alpha})$ is a diagonal matrix of p positive roughness parameter $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_p)$. Here $\boldsymbol{h} : \boldsymbol{X} \to \mathbb{R}^m$ is a vector of m regression functions $h_1(\boldsymbol{x}), ...h_m(\boldsymbol{x})$ and $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\{-(\boldsymbol{x}_i - \boldsymbol{x}_j)'[diag(\boldsymbol{\alpha})](\boldsymbol{x}_i - \boldsymbol{x}_j)\}, 1 \le i, j \le n$ with $c(\boldsymbol{x}, \boldsymbol{x}) = 1$.

Assume that $\boldsymbol{h}'(\boldsymbol{x}) = (1, \boldsymbol{x}')$, in which $m = p + 1$, and the joint prior distribution of $(\boldsymbol{B}, \boldsymbol{C})$ conditional on $\boldsymbol{\alpha}$ is Jeffreys-type non-informative prior ([14],[4]), which is independent of $\boldsymbol{\alpha}$,

$$\pi(\boldsymbol{B}, \boldsymbol{C}|\boldsymbol{\alpha}) \propto |\boldsymbol{C}|^{-(T+1)/2}.$$

The joint distribution of $\boldsymbol{Y}$ conditioning on hyper-parameters is multi-normal distribution

$$\boldsymbol{Y}|\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\alpha} \sim \mathcal{N}_{n,T}(\boldsymbol{HB}, \boldsymbol{A} \otimes \boldsymbol{C})$$

where $\boldsymbol{H}' = [\boldsymbol{h}(\boldsymbol{x}_1), ..., \boldsymbol{h}(\boldsymbol{x}_n)]$ and the between-input correlation matrix $\boldsymbol{A} = [c(\boldsymbol{x}_r, \boldsymbol{x}_l)]_{n\times n}$,

$1 \leq r, l \leq n$. The multi-output emulator is the posterior distribution of $\boldsymbol{y}(\cdot)$ given $\boldsymbol{Y}$. According to [4], the posterior distribution of $\boldsymbol{y}(\cdot)$ conditioning on $\boldsymbol{\alpha}$ and $\boldsymbol{Y}$ is the T-variate t distribution with $n-m$ degrees of freedom as follows

$$\boldsymbol{y}(\cdot)|\boldsymbol{\alpha}, \boldsymbol{Y} \sim \mathcal{T}(\boldsymbol{m}^*(\cdot), c^*(\cdot,\cdot)\hat{\boldsymbol{\Sigma}}_{GLS}; n-m) \tag{2.12}$$

in which

$$\boldsymbol{m}^*(\cdot) = \hat{\boldsymbol{B}}'_{GLS}\boldsymbol{h}(\cdot) + (\boldsymbol{Y} - \boldsymbol{H}\hat{\boldsymbol{B}}'_{GLS})'\boldsymbol{A}^{-1}\boldsymbol{t}(\cdot),$$

$$\begin{aligned} c^*(\boldsymbol{x}_i, \boldsymbol{x}_j) =& c(\boldsymbol{x}_i, \boldsymbol{x}_j) - \boldsymbol{t}'(\boldsymbol{x}_i)\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}_j) \\ &+ [\boldsymbol{h}(\boldsymbol{x}_i) - \boldsymbol{H}'\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}_i)]'(\boldsymbol{H}'\boldsymbol{A}^{-1}\boldsymbol{H})^{-1}[\boldsymbol{h}(\boldsymbol{x}_j) - \boldsymbol{H}'\boldsymbol{A}^{-1}\boldsymbol{t}(\boldsymbol{x}_j)] \end{aligned}$$

and $\hat{\boldsymbol{B}}_{GLS} = (\boldsymbol{H}'\boldsymbol{A}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}'\boldsymbol{A}^{-1}\boldsymbol{Y}$ is the generalized least squares estimator of $\boldsymbol{B}$, $\boldsymbol{t}(\cdot) = [c(\cdot, \boldsymbol{x}_1), ..., c(\cdot, \boldsymbol{x}_n)]'$.

To integrate (2.12) over $\boldsymbol{\alpha}$, according to Appendix A of [4], we have

$$\pi(\boldsymbol{\alpha}|\boldsymbol{Y}) \propto \pi(\boldsymbol{\alpha})|\boldsymbol{A}|^{-q/2}|\boldsymbol{H}'\boldsymbol{A}^{-1}\boldsymbol{H}|^{-t/2}|\boldsymbol{Y}'\boldsymbol{G}\boldsymbol{Y}|^{-(n-m)/2} \tag{2.13}$$

where $\pi(\boldsymbol{\alpha}) = \prod_{i=1}^{p}(1+\alpha_i^2)^{-1}$ is the Log-Logistic prior distribution of $\boldsymbol{\alpha}$ and $\boldsymbol{G} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{H}(\boldsymbol{H}'\boldsymbol{A}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}'\boldsymbol{A}^{-1}$. A Bayesian approach can be used to get the sampling from (2.13) in order to get the conditional posterior (2.12).

2.2.2 Singular Value Decomposition Based Gaussian Process Models with Quantitative Factors

In this section, we review the SVD based GP model introduced by [11]. Consider a T-dimensional output $\boldsymbol{y} = (y_1(\boldsymbol{x}), ...y_T(\boldsymbol{x}))$ with $n \times p$ design matrix as $\boldsymbol{X}$ given by

(2.4), and construct the emulator modelling the outputs using a p_w-dimensional basis representation [11]:

$$\boldsymbol{y} = \sum_{i=1}^{p_w} \boldsymbol{k}_i w_i(\boldsymbol{x}) + \boldsymbol{\varepsilon} \tag{2.14}$$

where $\{\boldsymbol{k}_1, ..., \boldsymbol{k}_{p_w}\}$ is a collection of principal components (PCs), which are orthogonal, T-dimensional basis vectors, $\{w_1(\boldsymbol{x}), ..., w_{p_w}(\boldsymbol{x})\}$ is a collection of PC weights, which are independent univariate GPs, and $\boldsymbol{\varepsilon}$ is a T-dimensional error term with distribution $\boldsymbol{\varepsilon} \sim N(0, \lambda_\varepsilon \boldsymbol{I}_T)$. Each PC weight $w_i(\boldsymbol{x})$ is treated as a realization of a mean zero GP

$$w_i(\boldsymbol{x}) \sim N(0, \lambda_{wi}^{-1} R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\rho}_{wi})) \tag{2.15}$$

where the covariance is defined by the correlation function $R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\rho}_{wi})$ with parameters $\boldsymbol{\rho}_{wi}$

$$R(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\rho}_{wi}) = \prod_{k=1}^{p} \rho_{wik}^{4(x_k - x'_k)^2}$$

and marginal precision λ_{wi}^{-1}.

Define $\boldsymbol{w}_i = (w_i(\boldsymbol{x}_1), ..., w_i(\boldsymbol{x}_n))'$, $i = 1, ..., p_w$. In addition, define $\boldsymbol{R}(\boldsymbol{X}; \boldsymbol{\rho}_{wi})$ to be the $n \times n$ correlation matrix with components as $[\boldsymbol{R}(\boldsymbol{X}; \boldsymbol{\rho}_{wi})]_{jk} = R((\boldsymbol{x}_j, \boldsymbol{x}_k); \boldsymbol{\rho}_{wi})$. Thus, the np_w vector $\boldsymbol{w} = (\boldsymbol{w}'_1, ..., \boldsymbol{w}'_{p_w})'$ has a multivariate normal distribution

$$\boldsymbol{w} \sim N(\boldsymbol{0}_{np_w}, \boldsymbol{\Sigma}_w), \tag{2.16}$$

which is controlled by a p_w vector $\boldsymbol{\lambda}_w = (\lambda_{w1}, ..., \lambda_{wp_w})$ and a pp_w vector of correlation parameters $\boldsymbol{\rho}_w = (\boldsymbol{\rho}'_{w1}, ..., \boldsymbol{\rho}'_{wp_w})'$. The covariance is

$$\boldsymbol{\Sigma}_w = \begin{pmatrix} \lambda_{w1}^{-1}\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\rho}_{w1}) & & \\ & \ddots & \\ & & \lambda_{wp_w}^{-1}\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\rho}_{wp_w}) \end{pmatrix}.$$

For training data $\boldsymbol{y}^n = (\boldsymbol{y}'_1, ..., \boldsymbol{y}'_n)'$, given the precision of error term as λ_ε, we have $\boldsymbol{y}^n = \boldsymbol{K}\boldsymbol{w} + \boldsymbol{\varepsilon}$ follows the distribution of

$$\boldsymbol{y}^n \sim N(\boldsymbol{0}_{nT}, \boldsymbol{K}\boldsymbol{\Sigma}_w\boldsymbol{K}' + \lambda_\varepsilon^{-1}\boldsymbol{I}_{nT}) \tag{2.17}$$

where $\boldsymbol{K}$ is an $nT \times np_w$ matrix given by $\boldsymbol{K} = [\boldsymbol{I}_n \otimes \boldsymbol{k}_1; \cdots; \boldsymbol{I}_n \otimes \boldsymbol{k}_{p_w}]$ with $\boldsymbol{k}_i$'s as PCs computed by SVD method. Thus, we have $\hat{\boldsymbol{w}} = (\boldsymbol{K}'\boldsymbol{K})^{-1}\boldsymbol{K}'\boldsymbol{y}^n$ and

$$\hat{\boldsymbol{w}} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{w}}} = \boldsymbol{\Sigma}_w + (\lambda_\varepsilon\boldsymbol{K}'\boldsymbol{K})^{-1}).$$

In practical situation, we need to estimate λ_ε, $\boldsymbol{\lambda}_w$ and $\boldsymbol{\rho}_w$ to build the emulators. [11] provides the estimates with Bayesian method. [11] selects Beta distribution prior for $\rho_{wik},\ i = 1, ..., p_w, k = 1, ..., p$ and Gamma distribution prior for λ_ε and $\lambda_{wi}, i = 1, ..., p_w$, i.e. $\rho_{wik} \sim Beta(a_{\rho_w}, b_{\rho_w})$, $\lambda_\varepsilon \sim Ga(a_\varepsilon, b_\varepsilon)$, $\lambda_{wi} \sim Ga(a_w, b_w)$,

$$\pi(\lambda_\varepsilon) \propto \lambda_\varepsilon^{a_\varepsilon - 1} e^{-b_\varepsilon\lambda_\varepsilon},$$

$$\pi(\lambda_{wi}) \propto \lambda_{wi}^{a_w - 1} e^{-b_w\lambda_{wi}}, i = 1, ..., p_w,$$

$$\pi(\rho_{wik}) \propto \rho_{wik}^{a_w - 1}(1 - \rho_{wik})^{b_w - 1}, i = 1, ..., p_w, k = 1, ..., p.$$

Referring to [11], the posterior distribution of the unknown parameters is

$$
\begin{aligned}
\pi(\lambda_\varepsilon, \boldsymbol{\lambda}_w, \boldsymbol{\rho}_w | \boldsymbol{y}^n) &\propto \pi(\lambda_\varepsilon, \boldsymbol{\lambda}_w, \boldsymbol{\rho}_w | \hat{\boldsymbol{w}}) \\
&\propto |\boldsymbol{\Sigma}_{\hat{w}}|^{-\frac{1}{2}} \times \exp[-\frac{1}{2}(\hat{\boldsymbol{w}})' \boldsymbol{\Sigma}_{\hat{w}} \hat{\boldsymbol{w}}] \times \lambda_\varepsilon^{a_\varepsilon - 1} e^{-b_\varepsilon \lambda_\varepsilon} \times \prod_{i=1}^{p_w} \lambda_{wi}^{a_w - 1} e^{-b_w \lambda_{wi}} \\
&\quad \times \prod_{i=1}^{p_w} \prod_{k=1}^{p} \rho_{wik}^{a_w - 1} (1 - \rho_{wik})^{b_w - 1}.
\end{aligned} \tag{2.18}
$$

We can draw the realizations of the unknown parameters using Metropolis–Hastings algorithm.

To predict the response at a new input $\boldsymbol{x}^*$ conditioning on parameters $\lambda_\varepsilon, \boldsymbol{\lambda}_w, \boldsymbol{\rho}_w$, we have the joint distribution of $(\boldsymbol{w}^*, \hat{\boldsymbol{w}})'$ as multivariate normal

$$
\begin{pmatrix} \boldsymbol{w}^* \\ \hat{\boldsymbol{w}} \end{pmatrix} \propto N\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} diag(\lambda_{wi}^{-1}; i = 1, ..., p_w) & \boldsymbol{\Sigma}_{w^*, \hat{w}} \\ \boldsymbol{\Sigma}'_{w^*, \hat{w}} & \boldsymbol{\Sigma}_{\hat{w}} \end{pmatrix} \right)
$$

where $\boldsymbol{\Sigma}_{w^*, \hat{w}}$ is the covariance matrix between w^* and $\hat{w}$. So the conditional mean of $\boldsymbol{w}^* = (w_1(\boldsymbol{x}^*), ..., w_{p_w}(\boldsymbol{x}^*))$ is

$$
\hat{\boldsymbol{w}}^* = E\{\boldsymbol{w}^* | \hat{\boldsymbol{w}}\} = \boldsymbol{\Sigma}_{w^*, \hat{w}} \boldsymbol{\Sigma}_{\hat{w}}^{-1} \hat{\boldsymbol{w}}.
$$

We can summarize Lemma 2.2.1 from Section 14.2 of [8] and use it to get the distribution of $\boldsymbol{y}(\boldsymbol{x}^*)$.

Lemma 2.2.1. *For* $(\boldsymbol{y} | \boldsymbol{\beta}, \sigma^2, \boldsymbol{X}) \sim N(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I})$ *and* $\boldsymbol{\beta} \sim N(\hat{\boldsymbol{\beta}}, \Sigma_\beta)$*, where* σ^2 *is the known covariance component. For a new observation* $\boldsymbol{y}_0$ *at input* $\boldsymbol{X}_0$*, we have*

$$
\boldsymbol{y}_0 | \boldsymbol{y}, \sigma^2 \sim N(\boldsymbol{X}_0 \hat{\beta}, \boldsymbol{X}_0 \Sigma_\beta \boldsymbol{X}_0' + \sigma^2 \boldsymbol{I}).
$$

Combing Lemma 2.2.1 and (2.17), we have the distribution of the predictive responses at an new input $\boldsymbol{x}^*$ as

$$\boldsymbol{y}(\boldsymbol{x}^*)|\boldsymbol{y}^n, \lambda_\varepsilon, \boldsymbol{\lambda}_w, \boldsymbol{\rho}_w \sim N(\sum_{i=1}^{p_w} \boldsymbol{k}_i \hat{w}_i(\boldsymbol{x}^*), \boldsymbol{x}^* \boldsymbol{\Sigma}_{\hat{w}}(\boldsymbol{x}^*)' + \lambda_\varepsilon^{-1} \boldsymbol{I}).$$

Another way to estimate the parameters λ_ε, $\boldsymbol{\lambda}_w$ and $\boldsymbol{\rho}_w$ is MLE method, which is used in [6]. In this case, the correlation function at two distinct inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are defined as

$$R(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\rho}_w) = \exp\{\sum_{k=1}^{p} (-\rho_{wk}(x_{1k} - x_{2k})^2)\} \tag{2.19}$$

controlled by correlation parameter $\boldsymbol{\rho}_w$. The MLE of λ_ε, $\boldsymbol{\lambda}_w$ and $\boldsymbol{\rho}_w$ is

$$(\hat{\lambda}_\varepsilon, \hat{\boldsymbol{\lambda}}_w, \hat{\boldsymbol{\rho}_w}) = argmax(L(\boldsymbol{y}^n|\lambda_\varepsilon, \boldsymbol{\lambda}_w, \boldsymbol{\rho}_w)).$$

2.2.3 Many Single Outputs Emulator with Quantitative Factors

We denote the computer model output as $y(\boldsymbol{x}, t)$ with input $\boldsymbol{x}$ at time t. Thus, with the inputs $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)'$, the corresponding output at time t is $((y(\boldsymbol{x}_1, t), ..., y(\boldsymbol{x}_n, t))'$. All outputs up to time T are denoted by

$$\boldsymbol{Y}(\boldsymbol{X}) = \begin{pmatrix} y(\boldsymbol{x}_1, 1) & \cdots & y(\boldsymbol{x}_1, T) \\ \vdots & \ddots & \vdots \\ y(\boldsymbol{x}_n, 1) & \cdots & y(\boldsymbol{x}_n, T) \end{pmatrix}_{n \times T}. \tag{2.20}$$

The predicted response at any new input $\boldsymbol{x}_0$ can be achieved separately at each time t, $1 \leq t \leq T$, each via a single output emulator with the corresponding column of matrix in (2.20). There are T emulators in total.

2.2.4 Time Input Emulator with Quantitative Factors

One simple way to deal with the problem of dynamic model is to extend the above by including time t as an input. Time input emulator is a single emulator, including time as an extra input, thus the input is a vector

$$\boldsymbol{X}_{TI} = [(\boldsymbol{x}_1, 1), ..., (\boldsymbol{x}_n, 1), (\boldsymbol{x}_1, 2), ..., (\boldsymbol{x}_n, 2), ..., (\boldsymbol{x}_1, T), ..., (\boldsymbol{x}_n, T)]' \tag{2.21}$$

and the corresponding output is an $nT \times 1$ vector

$$y_{TI} = [y(\boldsymbol{x}_1, 1), ..., y(\boldsymbol{x}_n, 1), y(\boldsymbol{x}_1, 2), ..., y(\boldsymbol{x}_n, 2), ..., y(\boldsymbol{x}_1, T), ..., y(\boldsymbol{x}_n, T)]'.$$

The prediction of any new input $(\boldsymbol{x}_0, t_0)$ can be achieved using (2.6).

2.2.5 Dynamic Linear Modelling with Gaussian Process for Quantitative Factors

In this section, we review the dynamic linear model with GP introduced by [17]. Consider the simulator with output $\boldsymbol{Y}$ and input $\boldsymbol{X}$. The output time series $\boldsymbol{Y}$ at design matrix $\boldsymbol{X}$ defined by (2.4) can be written as

$$\boldsymbol{Y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)' = \begin{pmatrix} Y(\boldsymbol{x}_1, 1) & \cdots & Y(\boldsymbol{x}_1, T) \\ \vdots & \ddots & \vdots \\ Y(\boldsymbol{x}_n, 1) & \cdots & Y(\boldsymbol{x}_n, T) \end{pmatrix}_{n \times T}$$

where $Y(\boldsymbol{x}_i, t), i = 1, ..., n, 1 \leq t \leq T$ represents the response of input $\boldsymbol{x}_i$ at time t. In addition, denote the i-th column of the above matrix is denoted as $\boldsymbol{Y}_i(\boldsymbol{X})$.

[17] proposed a model to build the relationship between the input space and output space utilizing GP models and time-varying autoregressive models (TVAR). For each $\boldsymbol{x}$ and t, we have a TVAR model

$$Y(\boldsymbol{x}, t) = \sum_{j=1}^{h} \psi_{t,j} Y(\boldsymbol{x}, t-j) + \varepsilon_t(\boldsymbol{x}),$$

where lag h is selected according to the data structure, for example in [17], h is selected to be 2; $\boldsymbol{\psi}_t = (\psi_{t,1}, ..., \psi_{t,h})'$ are unknown autoregressive parameters varying over time; $\varepsilon_t(\boldsymbol{x})$ is the residual varying over time. The covariance between outputs $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}$ at some fixed time t is written as

$$\mathrm{Cov}(\varepsilon_t(\boldsymbol{x}_i), \varepsilon_t(\boldsymbol{x}_j)) = \sigma_t^2 R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\theta})$$

with the same correlation functions $R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\theta})$ for all t and the variance σ_t^2 varies over time. In [17], the correlation function $R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\theta})$ is define by (2.3). Define $\boldsymbol{\varepsilon}_t = (\varepsilon_t(\boldsymbol{x}_1), \cdots, \varepsilon_t(\boldsymbol{x}_n))'$, the multivariate dynamic linear model can be presented as

$$\boldsymbol{Y}_t(\boldsymbol{X}) = \boldsymbol{F}_t' \boldsymbol{\psi}_t + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim N(0, \sigma_t^2 \boldsymbol{R})$$

where

$$\boldsymbol{F}_t' = \begin{pmatrix} y(\boldsymbol{x}_1, t-1) & \cdots & y(\boldsymbol{x}_1, t-h) \\ \vdots & \ddots & \vdots \\ y(\boldsymbol{x}_n, t-1) & \cdots & y(\boldsymbol{x}_n, t-h) \end{pmatrix}, \ \boldsymbol{R} = \begin{pmatrix} R(\boldsymbol{x}_1 - \boldsymbol{x}_1; \boldsymbol{\theta}) & \cdots & R(\boldsymbol{x}_1 - \boldsymbol{x}_n; \boldsymbol{\theta}) \\ \vdots & \ddots & \vdots \\ R(\boldsymbol{x}_n - \boldsymbol{x}_1; \boldsymbol{\theta}) & \cdots & R(\boldsymbol{x}_n - \boldsymbol{x}_n; \boldsymbol{\theta}) \end{pmatrix}$$

and $\boldsymbol{\psi}_t$ represents auto-regressive coefficients following a random walk

$$\boldsymbol{\psi}_t = \boldsymbol{\psi}_{t-1} + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim N(0, \boldsymbol{W}_t)$$

with initial value of $\boldsymbol{\psi}_0$, σ_t^2 and $\boldsymbol{W}_t$ specified by prior distributions.

The correlation function has two inputs $\boldsymbol{x}$ and t, we suppose that the correlation function can be decomposed as $R((\boldsymbol{x}_i, t_1), (\boldsymbol{x}_j, t_2)) = R(\boldsymbol{x}_i - \boldsymbol{x}_j; \boldsymbol{\theta}) \times R_{time}(|t_1 - t_2|; \boldsymbol{\theta}_{time})$, where $R_{time}(\cdot)$ is the correlation function on the time dimension. Thus, the GP model extended on time dimension, $\boldsymbol{Y}_{\mathcal{T}}(\boldsymbol{X}) = (\boldsymbol{Y}_1(\boldsymbol{X})', \cdots, \boldsymbol{Y}_T(\boldsymbol{X})')'$, has distribution $N_{nT}(\mu \times \mathbf{1}_{nT}, \boldsymbol{R} \otimes \boldsymbol{R}_{time}/\lambda)$, where λ controls the marginal precision of $\boldsymbol{Y}_{\mathcal{T}}(\boldsymbol{X})$. Omit the effect of $R_{time}(\cdot)$ after h lags, the distribution of $\boldsymbol{Y}_t(\boldsymbol{X})$ conditioning on the responses of $\boldsymbol{X}$ at all the previous time points can be written as

$$\begin{aligned}
&\pi(\boldsymbol{Y}_t(\boldsymbol{X})|\boldsymbol{Y}_{t-1}(\boldsymbol{X}), ..., \boldsymbol{Y}_1(\boldsymbol{X}), \boldsymbol{\theta}, \boldsymbol{\theta}_{time}) \\
&\qquad = \pi(\boldsymbol{Y}_t(\boldsymbol{X})|\boldsymbol{Y}_{t-1}(\boldsymbol{X}), ..., \boldsymbol{y}_{t-h}(\boldsymbol{X}), \boldsymbol{\theta}, \boldsymbol{\theta}_{time}), \ t > h
\end{aligned}$$

in which $(\boldsymbol{\theta}, \boldsymbol{\theta}_{time})$ is the hyper-parameter controlling the covariance function. The conditional distribution is multivariate normal with mean and variance as

$$E(\boldsymbol{Y}_t(\boldsymbol{X})|\boldsymbol{Y}_{t-1}(\boldsymbol{X}), ..., \boldsymbol{y}_{t-h}(\boldsymbol{X}), \boldsymbol{\theta}, \boldsymbol{\theta}_{time}) = \boldsymbol{F}_t'\boldsymbol{\psi}_t \ \text{ with } \ \boldsymbol{\psi}_t = \tilde{\boldsymbol{R}}_{time,h}^{-1}\boldsymbol{\rho}_{t,t-1:t-h}$$

and

$$\text{Var}\,(\boldsymbol{Y}_t(\boldsymbol{X})|\boldsymbol{Y}_{t-1}(\boldsymbol{X}), ..., \boldsymbol{y}_{t-h}(\boldsymbol{X}), \boldsymbol{\theta}, \boldsymbol{\theta}_{time}) = \frac{1}{\lambda}(1 - \boldsymbol{\rho}_{t,t-1:t-h}'\tilde{\boldsymbol{R}}_{time,h}^{-1}\boldsymbol{\rho}_{t,t-1:t-h})\boldsymbol{R}$$

where $\boldsymbol{\rho}_{t,t-1:t-h} = (R_{time}(t-(t-1)), ..., R_{time}(t-(t-h)))' = (R_{time}(1), ..., R_{time}(h))'$

and $\tilde{\boldsymbol{R}}_{time,h}^{-1}$ is a $h \times h$ matrix with (i,j) element $(\tilde{\boldsymbol{R}}_{time,h}^{-1})_{i,j} = R_{time}(|i-j|;\boldsymbol{\theta}_{time})$.

In the analysis of [17], all the correlation parameters α_k's in (2.3) is fixed to be 2. MCMC methodology with Metropolis-Hasting scheme is used to simulate the joint posterior distribution of $\boldsymbol{\psi}_{1:T} = \{\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_T\}$, $\boldsymbol{\sigma}_{1:T}^2 = (\sigma_1^2, ...\sigma_T^2)$ and range parameters $\boldsymbol{\theta}_{1:p} = (\theta_1, \cdots, \theta_p)$ in (2.3). According to [17], the joint posterior distribution is

$$\pi(\boldsymbol{\psi}_{1:T}, \boldsymbol{\sigma}_{1:T}, \boldsymbol{\theta}_{1:p}|\boldsymbol{Y}) \propto \pi(\boldsymbol{Y}|\boldsymbol{\psi}_{1:T}, \boldsymbol{\sigma}_{1:T}, \boldsymbol{\theta}_{1:p}) \times \pi(\boldsymbol{\theta}_{1:p}) \times \prod_{t=1}^{T} \pi(\boldsymbol{\psi}_t, \boldsymbol{\sigma}_t|\boldsymbol{y}_1, ..., \boldsymbol{y}_{t-1}, \boldsymbol{\theta}_{1:p})$$

with the conditional distribution of $\boldsymbol{Y}$ is

$$\pi(\boldsymbol{Y}|\boldsymbol{\psi}_{1:T}, \boldsymbol{\sigma}_{1:T}, \boldsymbol{\theta}_{1:p}) \propto |\boldsymbol{R}|^{-(T-h)/2} \exp(-Q/2) \prod_{t=h+1}^{T} (\sigma_t^2)^{-n/2}$$

where

$$Q = \sum_{t=h+1}^{T} \frac{1}{\sigma_t^2} (\boldsymbol{Y}_t(\boldsymbol{X}) - \boldsymbol{F}_t\boldsymbol{\psi}_t)' \boldsymbol{R}^{-1} (\boldsymbol{Y}_t(\boldsymbol{X}) - \boldsymbol{F}_t\boldsymbol{\psi}_t).$$

To estimate the parameters, [17] uses block Gibbs samplers with a Metropolis-within-Gibbs step for $\theta_1, \ldots, \theta_p$. At each Gibbs step g, $\boldsymbol{\theta}_{1:p}^{(g)}$ is drawn from $\pi(\boldsymbol{\theta}_{1:p}|\boldsymbol{Y}, \boldsymbol{\psi}_{1:T}^{(g-1)}$, $(\boldsymbol{\sigma}_{1:T}^2)^{(g-1)})$ via Metropolis-Hastings algorithm, and then sample $\pi(\boldsymbol{\psi}_{1:T}, \boldsymbol{\sigma}_{1:T}^2|\boldsymbol{Y}, \boldsymbol{\theta}_{1:p}^{(g)})$ via the forward filtering and backward sampling algorithm. The initial prior of $\boldsymbol{\psi}_0$ follows normal distribution and σ_0^2 follows gamma distribution with some initial parameters detailed explained in Appendix 2 of [17].

The prediction at any new input $\boldsymbol{x}_0$ at time t is normal with mean

$$\boldsymbol{\mu}_t(\boldsymbol{x}_0) = \sum_{j=1}^{h} \psi_{t,j} Y(\boldsymbol{x}_0, t-j) + \frac{1}{\sigma_t^2} \boldsymbol{r}_{\boldsymbol{x}_0}' \boldsymbol{R}^{-1} \boldsymbol{\varepsilon}_t$$

and variance

$$\sigma_t^2(\boldsymbol{x}_0) = \sigma_t^2(1 - \boldsymbol{r}'_{\boldsymbol{x}_0}\boldsymbol{R}^{-1}\boldsymbol{r}_{\boldsymbol{x}_0})$$

where $\boldsymbol{r}_{\boldsymbol{x}_0} = R((\boldsymbol{x}_0, \boldsymbol{x}_1), ..., R(\boldsymbol{x}_0, \boldsymbol{x}_n))'$, $\boldsymbol{\varepsilon}_t = (\varepsilon_t(\boldsymbol{x}_1), ..., \varepsilon_t(\boldsymbol{x}_n))'$ and $\varepsilon_t(\boldsymbol{x}_i) = Y(\boldsymbol{x}_i, t) - \sum_{j=1}^{h} \psi_{t,j} Y(\boldsymbol{x}_i, t-j)$.

2.3 Additive Gaussian Process Model with Quantitative and Qualitative Variables

In this section we review the additive GP model with quantitative and qualitative factors firstly introduced by [26]. Consider a computer model with input $\boldsymbol{w} = (\boldsymbol{x}', \boldsymbol{z}')'$, where $\boldsymbol{x} = (x_1, ..., x_p)'$ is a p-dimensional vector of quantitative variables and $\boldsymbol{z} = (z_1, ..., z_q)'$ is a q-dimensional vector of qualitative variables in the input, and each element z_j in $\boldsymbol{z}$ has m_j levels. The response of the computer model can be represented as

$$y(\boldsymbol{w}) = \boldsymbol{f}'(\boldsymbol{w})\boldsymbol{\beta} + \varepsilon(\boldsymbol{w}) \tag{2.22}$$

where the $\boldsymbol{f}(\boldsymbol{w}) = (f_1(\boldsymbol{w}), ..., f_m(\boldsymbol{w}))'$ are specified regression functions, $\boldsymbol{\beta} = (\beta_1, ..., \beta_m)'$ is an $m \times 1$ unknown regression coefficient, and $\varepsilon(\cdot)$ is a zero mean GP with variance σ_ε^2 and some correlation function $R_w(\cdot)$. Because of the change of the structure of input variable, correlation function (2.3) cannot describe the relationship between inputs, so new correlation structure is necessary. One method to build correlation function $R_w(\cdot)$ for the dataset with both quantitative and qualitative factors is the additive GP model [26].

To simplify calculation, we model the response y as

$$y = \mu + \varepsilon(\boldsymbol{w}), \quad \varepsilon(\boldsymbol{w}) = G_1(\boldsymbol{x}, z_1) + \cdots + G_q(\boldsymbol{x}, z_q)$$

where μ is the overall mean and G_j is mean zero GP. For each qualitative factor z_j, we define an $m_j \times m_j$ correlation matrix $\boldsymbol{T}^{(j)} = [\tau^{(j)}_{z_{rj}, z_{sj}}], 1 \leq r, s \leq m_j, 1 \leq j \leq q$. Thus, the covariance function of GP G_j is defined as

$$\operatorname{Cov}_{G_j}(\boldsymbol{w}_1, \boldsymbol{w}_2) = \sigma_j^2 \tau^{(j)}_{z_{1j}, z_{2j}} R(\boldsymbol{x}_1, \boldsymbol{x}_2 | \boldsymbol{\theta}^{(j)}) \tag{2.23}$$

where σ_j^2 is the covariance component of qualitative factor z_j, $\tau^{(j)}_{z_{1j}, z_{2j}}$ represents the correlation of qualitative factor z_j and $R(\cdot)$ is the correlation function of quantitive factors, one common choice is (2.3). Thus, the response $y(\boldsymbol{w})$ that follows additive GP has covariance function

$$\operatorname{Cov}(y(\boldsymbol{w}_1), y(\boldsymbol{w}_2)) = \sum_{j=1}^{q} \sigma_j^2 \tau^{(j)}_{z_{1j}, z_{2j}} R(\boldsymbol{x}_1, \boldsymbol{x}_2 | \boldsymbol{\theta}^{(j)}).$$

Also, we define the covariance matrix of $\boldsymbol{y}$ as $\Sigma = [\operatorname{Cov}(y(\boldsymbol{w}_i), y(\boldsymbol{w}_j))]_{n \times n}$ controlled by parameters $(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{T})$.

The parameters needed to be estimated in the model are μ, $\boldsymbol{\sigma}^2 = (\sigma_1^2, ..., \sigma_q^2)$, $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(q)})$ and $\boldsymbol{T} = \{\boldsymbol{T}^{(1)}, ..., \boldsymbol{T}^{(q)}\}$. To build the correlation matrix $\boldsymbol{T}$ for qualitative factors, [26] uses the hypersphere decomposition which was initially employed in [25]. As known from [21], $\boldsymbol{T}^{(1)}, ..., \boldsymbol{T}^{(q)}$ should be q positive definite matrix with unit diagonal elements. Take $\boldsymbol{T}^{(j)}$ as an example, we have $\boldsymbol{T}^{(j)} = L_j L_j'$ where $L_j = (l^{(j)}_{r,s})$ is a Cholesky decomposition of $\boldsymbol{T}^{(j)}$. Each row vector $(l^{(j)}_{r,1}, ..., l^{(j)}_{r,r})$ in L_j is specified by

- $r = 1$, $l^{(j)}_{r,r} = 1$

- $r = 2, ..., m_j$,

$$\begin{cases} l^{(j)}_{r,1} = \cos(\varphi^{(j)}_{r,1}) \\ l^{(j)}_{r,s} = \sin(\varphi^{(j)}_{r,1}) \cdots \sin(\varphi^{(j)}_{r,s-1}) \cos(\varphi^{(j)}_{r,s}), s = 2, ..., r-1 \\ l^{(j)}_{r,r} = \sin(\varphi^{(j)}_{r,1}) \cdots \sin(\varphi^{(j)}_{r,r-1}) \end{cases} \tag{2.24}$$

where $\varphi^{(j)}_{i,k} \in (0, \pi), 1 \leq k \leq i, 1 \leq i \leq m_j, 1 \leq j \leq q$. In addition, we have

$$\tau_{r,r} = \sum_{s=1}^{r} l^2_{r,s} = 1$$

which assure the diagonal element of $\boldsymbol{T}^{(j)}$ is 1. Thus, $\boldsymbol{T}$ is controlled by parameters $\varphi^{(j)}_{i,k}, 1 \leq k \leq i, 1 \leq i \leq m_j, 1 \leq j \leq q$. The log-likelihood function is

$$l(\mu, \boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{T}|\boldsymbol{y}) \propto -\frac{1}{2}(\ln|\boldsymbol{\Sigma}| + (\boldsymbol{y} - \mu\mathbf{1})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \mu\mathbf{1})). \tag{2.25}$$

Given $(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{T})$, we can get the MLE of μ

$$\hat{\mu} = (\mathbf{1}'\Sigma^{-1}\mathbf{1})^{-1}\mathbf{1}'\Sigma^{-1}\boldsymbol{y}. \tag{2.26}$$

Substituting (2.26) to (2.25), we can obtain [26]

$$l(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{T}|\boldsymbol{y}) \propto -\frac{1}{2}(\ln|\boldsymbol{\Sigma}| + \boldsymbol{y}'\Sigma^{-1}\boldsymbol{y} - (\mathbf{1}'\Sigma^{-1}\mathbf{1})^{-1}(\mathbf{1}'\Sigma^{-1}\boldsymbol{y})^2).$$

The MLE of $(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{T})$ can be approached by

$$(\hat{\boldsymbol{\sigma}}^2, \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{T}}) = \text{argmax} \ \ l(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{T} | \boldsymbol{y}).$$

The prediction of $y(\boldsymbol{w}_0)$ at a new input $\boldsymbol{w}_0$ is

$$\hat{y}(\boldsymbol{w}) = \hat{\mu} + \Sigma_0' \Sigma^{-1} (\boldsymbol{y} - \hat{\mu}\mathbf{1})$$

where $\Sigma_0 = (\text{Cov}\,(y(\boldsymbol{w}_0), y(\boldsymbol{w}_1)), \ldots, \text{Cov}\,(y(\boldsymbol{w}_0), y(\boldsymbol{w}_n)))'$. Note that $(\boldsymbol{\sigma}^2, \boldsymbol{\theta}, \boldsymbol{T})$ are embedded in Σ and Σ_0.

Chapter 3

Comparison of Models with Only Quantitative Factors

In this section, we conduct the comparison of four models with quantitative inputs reviewed in Section 2.2.2-2.2.5 for three examples with different simulators. The four models are many single outputs (MS) model, time input (TI) model, SVD based GP (SVD) model and dynamic linear model (DLM).

The indicator to show the performance of different models is root mean square prediction error (RMSPE) [24], which is

$$RMSPE = \sqrt{\frac{1}{|\boldsymbol{X}_{test}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{test}} (\{\boldsymbol{y}(\boldsymbol{x}) - \hat{\boldsymbol{y}}(\boldsymbol{x}))^2\}} \tag{3.1}$$

where $\boldsymbol{X}_{test}$ is the testing data set, $|\boldsymbol{X}_{test}|$ is the size of the testing data set, $\boldsymbol{y}(\boldsymbol{x})$ and $\hat{\boldsymbol{y}}(\boldsymbol{x})$ are the $T \times 1$ vectors of the predicted responses and true responses at new input $\boldsymbol{x}$. We use maximin Latin hypercube design to generate the training data and testing data. For each example, we calculate the RMSPEs in (3.1) to evaluate the effectiveness of MS model, TI model, SVD based GP model and DLM over 50 replications. MS, TI, SVD and DLM models are used in Example 1. TI model is not applicable in Examples 2-3 because the dataset with long time-series outputs is computationally demanding for TI model. The design matrix of TI model is a 20000×4 matrix in Example 2 and a 20000×5 matrix in Example 3. Thus, MS, only SVD and DLM are tested in Examples 2-3.

3.1 Example 1

Consider the first example with 6-dimensional inputs and time-series outputs. Data is generated from the simulator [20]

$$y(\boldsymbol{x}, t) = l(x_1, x_2) + l(x_1, t) + l(x_4, x_5) + 0.3l(x_6, t), \tag{3.2}$$

where $l(\cdot, \cdot)$ is the Langermann function defined as

$$l(x_1, x_2) = c \exp(-\frac{1}{\pi}\sum_{j=1}^{2}(x_j - A_j)^2) cos(\pi \sum_{j=1}^{2}(x_j - A_j)^2)),$$

$A = (3, 5)$, $c = 0.1$, $\boldsymbol{x} \in (0, 2) \times (0, 4) \times (0, 1) \times (0, 1) \times (0, 0.75) \times (0, 0.75)$ and time points are 50 equally spaced points in the range $t \in [1.5, 3.5]$. The Langermann function is a multimodal test function and it is non-symmetric with its local minima unevenly distributed [13]. For each of the 50 replications, we generated the training data of size 100 with maximin Latin hypercube design while we generate test data of size 1000 with random Latin hypercube design. Figure 3.1 displays the boxplot of the RMSPEs associated with the MS model, the SVD model and DLM model over 50

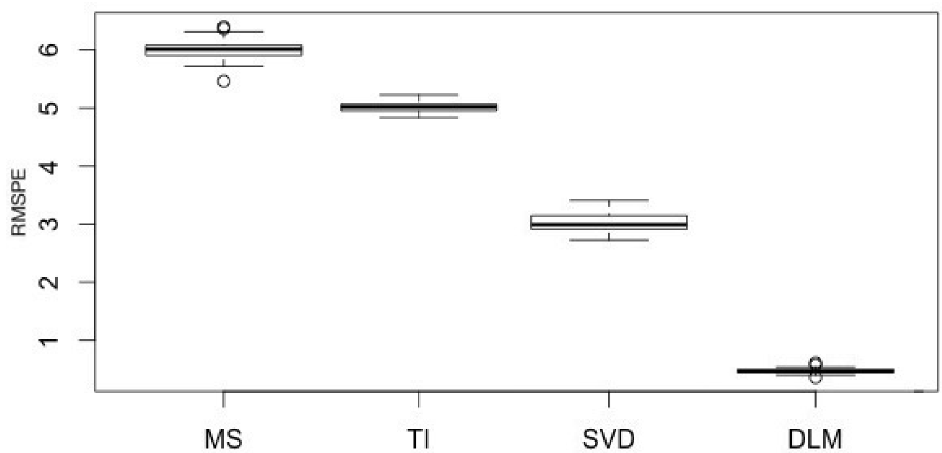

Figure 3.1: The boxplot of the RMSPEs computed from simulator (3.2) of MS, TI, SVD and DLM models over 50 replications.

simulation runs. The results of Figure 3.1 indicate that the DLM model outperforms the MS, TI and SVD models.

3.2 Example 2

Consider the second example with 3-dimensional inputs and time-series outputs. Data is generated from the simulator [20]

$$y(\boldsymbol{x}, t) = m(x_1, x_2) + 2 * m(x_3, t) + 0.75 * m(x_1 t, x_3), \tag{3.3}$$

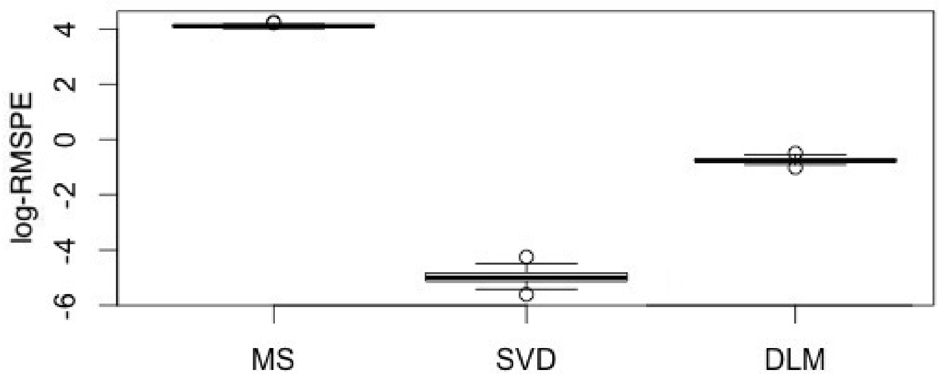

Figure 3.2: The boxplot of the log-RMSPEs computed from simulator (3.3) of MS, SVD and DLM models over 50 replications.

where $m(\cdot, \cdot)$ is the Mccormick function defined as

$$m(x_1, x_2) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1, \tag{3.4}$$

$\boldsymbol{x} \in (-1, 2) \times (-3, 3) \times (-1, 1)$ and time points are 200 equally spaced points in the range $t \in [1, 2]$. The Mccormick function is a plated-shape, continuous and convex function to evaluate the performance of algorithms [12]. For each of the 50 replications, we generated the training data of size 100 with maximin Latin hypercube design while we generate test data of size 1000 with random Latin hypercube design. From the results in Figure 3.2, it is clear that the SVD model gives a smaller RMSPEs than the MS and DLM models.

3.3 Example 3

Consider an example is to estimate pollutant concentration in a chemical accident [2]. This simulator has 5-dimensional inputs and time-series outputs. Data is generated from the simulator

$$f(\boldsymbol{x}, t) = \frac{M}{4\pi Dt} \exp\{\frac{-s^2}{4Dt}\} + \frac{M}{4\pi D(t-\tau)} \exp\{\frac{-(s-L)^2}{4D(t-\tau)}\} \cdot I(\tau < t), \tag{3.5}$$

where $\boldsymbol{x} = (M, D, L, \tau, s)$, M is the mass spilled at each location, D is the diffusion rate, L and τ are the location and the time of the second spill, and s is the spatial location, $\boldsymbol{x} \in [7, 13] \times [0.02, 0.12] \times [0.01, 3] \times [30.01, 30.295] \times [0, 3]$ and time points are 200 equal spaced points in the range $t \in [0.3, 60]$. For each of the 50 replications, we generated the training data of size 100 with maximin Latin hypercube design while we generate test data of size 1000 with random Latin hypercube design. Figure 3.3 displays the boxplot of the RMSPEs associated with MS, SVD and DLM models over 50 simulation runs and it indicates that the SVD model is better than the MS and DLM models the same as in Example 2.

From the three examples in the simulation study of Section 3.2, it is clear that the DLM model and the SVD model outperform with regard to the indicator root mean square prediction error. Thus, in the following study, we try to build the model with quantitative and qualitative factors based on the DLM model and the SVD model.

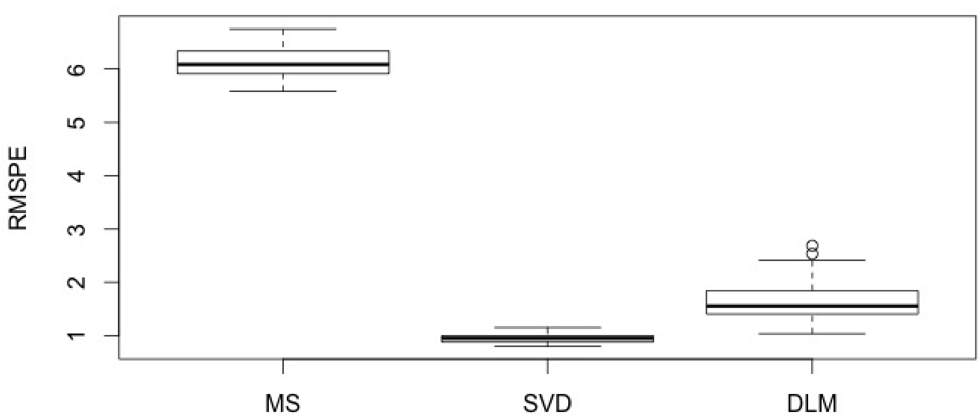

Figure 3.3: The boxplot of the RMSPEs computed from simulator (3.5) of MS, SVD and DLM models over 50 replications.

Chapter 4

Modelling Approaches with Both Quantitative and Qualitative Factors

This chapter introduces two models with both quantitative and qualitative factors. We incorporate the qualitative factors into DLM model and SVD model respectively. To incorporate the qualitative factors, we use the additive GP model which we review in Section 2.3. The additive GP model is advantageous as it assigns a covariance component to each qualitative factors. If there is no correlation for some qualitative factor, the correlation for quantitative and other qualitative factors can still be presented. As shown in the simulation study in Chapter 3, DLM model and SVD model provide more accurate predictions, we use these two models to incorporate qualitative factors in this chapter. Section 4.1 introduces the dynamic linear model with additive GP (DLM-ADGP) for time-series outputs and inputs with quantitative and qualitative factors. To obtain the parameters in the model, we use Bayesian method and Markov Chain Monte Carlo (MCMC) methodology to estimate the parameters from the corresponding posterior distributions. Section 4.2 describes the SVD based additive GP model (SVD-ADGP). In this model, we use the SVD method to deal with the computational problem of long time-series and use the additive GP model to incorporate qualitative factors. We use bayesian approach and maximum a posterior (MAP) method to estimate the unknown parameters in the model.

4.1 Dynamic Linear Model with Additive Gaussian Process

4.1.1 Model Description

Consider a computer experiment with p quantitative input variables $\boldsymbol{x} = (x_1, \ldots, x_p)$, q qualitative input variables $\boldsymbol{z} = (z_1, \ldots, z_q)$ and times series outputs. Let the input be $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z})$, and t represents the time point, $1 \le t \le T$. Assume that each qualitative factor z_j has m_j levels. For each time point, there are n responses. The response at $(\boldsymbol{w}, t)$ is $y(\boldsymbol{w}, t)$. The design matrix $\boldsymbol{D}$ is

$$\boldsymbol{D} = \begin{pmatrix} (\boldsymbol{w}_1, 1) & \cdots & (\boldsymbol{w}_1, T) \\ \vdots & \ddots & \vdots \\ (\boldsymbol{w}_n, 1) & \cdots & (\boldsymbol{w}_n, T) \end{pmatrix}.$$

We build the relationship between the input space and output space utilizing GP models and time-varying autoregressive models. For each input $\boldsymbol{w}$ at time t, we have a time-varying autoregressive model

$$y(\boldsymbol{w}, t) = \sum_{j=1}^{h} \psi_{t,j} y(\boldsymbol{w}, t-j) + \varepsilon_t(\boldsymbol{w}). \tag{4.1}$$

With notation of outputs $\boldsymbol{y}_t = (y(\boldsymbol{w}_1, t), ..., y(\boldsymbol{w}_n, t))'$, auto-regressive coefficients $\boldsymbol{\psi}_t = (\psi_{t,1}, ..., \psi_{t,h})'$ and residuals $\boldsymbol{\varepsilon}_t = (\varepsilon_t(\boldsymbol{w}_1), ..., \varepsilon_t(\boldsymbol{w}_n))'$, $\boldsymbol{\varepsilon}_t \sim N(0, \Phi_t)$, we can write the dynamic linear model with additive GPs (DLM-ADGP model) over all inputs as

$$\boldsymbol{y}_t = \boldsymbol{F}_t' \boldsymbol{\psi}_t + \boldsymbol{\varepsilon}_t \tag{4.2}$$

where

$$\boldsymbol{F}_t' = \begin{pmatrix} y(\boldsymbol{w}_1, t-1) & \cdots & y(\boldsymbol{w}_1, t-h) \\ \vdots & \ddots & \vdots \\ y(\boldsymbol{w}_n, t-1) & \cdots & y(\boldsymbol{w}_n, t-h) \end{pmatrix}.$$

To describe the correlation structure of qualitative factors, we regard $\varepsilon_t(\boldsymbol{w})$ as an additive GP as

$$\varepsilon_t(\boldsymbol{w}) = \sqrt{\nu_t}(G_1(\boldsymbol{x}, z_1) + ... + G_q(\boldsymbol{x}, z_q)),$$

where $G_i \sim GP(0, \sigma_i^2)$ and ν_t represents the variance effect of different time. Denote $\boldsymbol{T}^{(j)} = (\tau_{r,s}^{(j)})$ as the $m_j \times m_j$ correlation matrix, with m_j as the number of levels of the j-th qualitative factor z_j, which is a positive-definite matrix with unit diagonal elements with a Cholesky-type decomposition $\boldsymbol{T}^{(j)} = \boldsymbol{L}_j \boldsymbol{L}_j'$, $\boldsymbol{L}_j$ is defined as (2.24). Thus, the covariance between two inputs $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ is

$$\phi(y(\boldsymbol{w}_1), y(\boldsymbol{w}_2)) = \nu_t \sum_{j=1}^{q} \sigma_j^2 \tau_{z_{1j}, z_{2j}}^{(j)} R(\boldsymbol{x}_1, \boldsymbol{x}_2 | \boldsymbol{\theta}^{(j)}).$$

We denote the $n \times n$ covariance matrix $\mathbf{\Phi}_t$ as $\mathbf{\Phi}_t = [\phi(y(\boldsymbol{w}_i), y(\boldsymbol{w}_{i'}))]_{n\times n}$ controlled by parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(q)})$, $\boldsymbol{\sigma}^2 = (\sigma_1^2, ..., \sigma_q^2)$, $\boldsymbol{T} = (\boldsymbol{T}^{(1)}, ..., \boldsymbol{T}^{(q)})$. The covariance component of Φ_t is $v_t = \nu_t \sum_{i=1}^q \sigma_i^2$, the entries of the correlation matrix $\boldsymbol{R}$ is $\boldsymbol{R}_{i,k} = \frac{1}{\sum_{j=1}^q \sigma_j^2} \sum_{j=1}^q \sigma_j^2 \tau_{z_{1j},z_{2j}}^{(j)} R(\boldsymbol{x}_i, \boldsymbol{x}_k|\boldsymbol{\theta}^{(j)})$, where $R(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{\theta}^{(j)})$ describes the correlation between quantitative factors. In this model, we use the most popular one

$$R(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{\theta}^{(j)}) = \exp(-\sum_{k=1}^{p} \theta_k^{(j)} |x_{1k} - x_{2k}|^2).$$

Also, we assume the auto-regressive coefficients follow a random walk over time

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

where $\boldsymbol{W}_t$ is the time-varying parameter with specified prior.

4.1.2 Parameters Estimation

In the DLM-ADGP model, the parameters need to be estimated are $\boldsymbol{\psi}, \boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\varphi}$. $\boldsymbol{\psi} = \{\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_T\}$ where $\boldsymbol{\psi}_i = (\psi_{i1}, ..., \psi_{ih})$, $\boldsymbol{v} = \{v_1, ..., v_T\}$, $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(q)}\}$ where $\boldsymbol{\theta}^{(i)} = (\theta_1^{(i)}, ..., \theta_p^{(i)})$, $\boldsymbol{\sigma}^2 = (\sigma_1^2, ..., \sigma_q^2)$,$\boldsymbol{\varphi} = \{\boldsymbol{\varphi}^{(1)}, ..., \boldsymbol{\varphi}^{(q)}\}$, and parameters in $\boldsymbol{T}^{(j)}$ are $\boldsymbol{\varphi}^{(j)} = \{\boldsymbol{\varphi}_2^{(j)}, ..., \boldsymbol{\varphi}_{m_j}^{(j)}\}$, $\boldsymbol{\varphi}_r^{(j)} = (\varphi_{r,1}^{(j)}, ..., \varphi_{r,r-1}^{(j)})'$ where $2 \leq r \leq m_j$, $1 \leq j \leq q$ defined as (2.24). We use the Bayesian method to approach the parameters. Referring to

[17], the conditional likelihood function proportional to

$$p(\boldsymbol{y}|\psi, \boldsymbol{v},\boldsymbol{\theta}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi}) \propto \prod_{t=p+1}^{T} |\boldsymbol{\Phi}_t|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{\Phi}_t^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t))$$
$$\propto |\boldsymbol{R}|^{-\frac{T-p}{2}} \times \prod_{t=p+1}^{T} (v_t^{-n/2}) \times \exp(-\frac{1}{2}\sum_{t=p+1}^{T}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{R}^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)/v_t) \tag{4.3}$$

where $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{\theta}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi})$ is the correlation matrix and $\boldsymbol{\Phi}_t = v_t\boldsymbol{R}$ is the covariance matrix at time t. Thus, the posterior distribution of unknown parameters is

$$p(\boldsymbol{\psi}, \boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\psi}, \boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi}) \times \prod_{t=1}^{T} p(\boldsymbol{\psi}_t, v_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi})$$
$$\times\, p(\boldsymbol{\theta}) \times p(\boldsymbol{\sigma^2}) \times p(\boldsymbol{\varphi}),$$

and for $t = 1$, we define $\boldsymbol{y}_{1:0} = 1$ in $p(\boldsymbol{\psi}_t, v_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi})$.

For the Bayesian method, the prior distribution of parameters need to be chosen carefully. For the range parameter $\boldsymbol{\theta}$, according to [11] and [17], we first define $\rho_i^{(j)} = \exp(-\theta_i^{(j)}/4)$, then set the prior distribution of $\rho_i^{(j)}$ as Beta distribution $Beta(a_0, b_0)$. Thus, the joint prior distribution of $\boldsymbol{\theta}$ is

$$p(\boldsymbol{\theta}) \propto \prod_{j=1}^{q}\prod_{i=1}^{p}(1 - \exp(-\theta_i^{(j)}/4))^{b-1} \exp(-\theta_i^{(j)}/4)^{a-1}. \tag{4.4}$$

For the range parameter $\boldsymbol{\sigma}^2$, we choose the commonly used inverse gamma distribution for each element, that is

$$\sigma_i^2 \sim IG(\alpha_i^{(0)}, \beta_i^{(0)}).$$

The prior joint distribution of $\boldsymbol{\sigma}^2$ is

$$p(\boldsymbol{\sigma^2}) \propto \prod_{i=1}^{q} (\sigma^2)^{-\alpha_i^{(0)}-1} e^{-\beta_i^{(0)}\sigma_i^2} \tag{4.5}$$

where $\alpha_i^{(0)}, \beta_i^{(0)}$ are specified prior. We treat the parameters in $\boldsymbol{T}^{(j)}$ as $\boldsymbol{\varphi}^{(j)} = \{\boldsymbol{\varphi}_2^{(j)}, ..., \boldsymbol{\varphi}_{m_j}^{(j)}\}$, $\boldsymbol{\varphi}_r^{(j)} = (\varphi_{r,1}^{(j)}, ..., \varphi_{r,r-1}^{(j)})'$ where $2 \leq r \leq m_j$ according to (2.24). All elements in $\boldsymbol{\varphi}$ are mutually independent. We assume the prior distribution is $\varphi_{r,s}^{(j)} \sim Unif(0, \pi)$, and $p(\varphi_{r,s}^{(j)}) = \frac{1}{\pi}$, so the prior distribution of $\boldsymbol{\varphi}$ can be written as

$$\begin{aligned} p(\boldsymbol{\varphi}) &= \prod_{j=1}^{q} p(\boldsymbol{\varphi}^{(j)}) \\ p(\boldsymbol{\varphi}^{(j)}) &= \prod_{r=2}^{m_j} \prod_{s=1}^{r-1} p(\varphi_{r,s}^{(j)}) = (\frac{1}{\pi})^{(r-1)(m_j-1)}. \end{aligned} \tag{4.6}$$

For the priors of auto-regressive coefficients $\boldsymbol{\psi}_0$ and covariance component v_0, we assume the prior distributions [17] are

$$(\boldsymbol{\psi}_0|v_0, D_0) \sim N(\boldsymbol{m}_0, v_0\boldsymbol{C}_0), \quad v_0^{-1}|D_0 \sim Ga(n_0/2, d_0/2),$$

where D_0 is the initial prior setting for prior setting of all parameters. In this case, the posterior distribution for $(\boldsymbol{\psi}_{t-1}, v_{t-1}|D_{t-1})$ is normal/inverse Gamma distribution.

We use Gibbs samplers with a Metropolis-within-Gibbs step [17] for $\boldsymbol{\psi}, \boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{\sigma}^2, \boldsymbol{\varphi}$. At the g-th Gibbs step, we need to sample the parameters from the corresponding full conditional posterior distribution. Firstly, we sample $(\boldsymbol{\psi}_{(g)}, \boldsymbol{v}_{(g)})$ from normal/inverse Gamma distribution $p(\boldsymbol{\psi}_{(g)}, \boldsymbol{v}_{(g)}|\boldsymbol{y}, \boldsymbol{\theta}_{(g-1)}, (\boldsymbol{\sigma^2})_{(g-1)}, \boldsymbol{\varphi}_{(g-1)})$ shown in (4.7) and (4.8) by the forward filtering and backward sampling algorithm. Secondly, we sample $\boldsymbol{\theta}_{(g)}$,

$(\boldsymbol{\sigma^2})_{(g)}$ and $\boldsymbol{\varphi}_{(g)}$ from their full conditional posterior distribution (4.9), (4.10) and (4.11) in sequence utilizing the Metropolis-Hastings algorithm.

In each Gibbs step, the forward filtering method is used to achieve the estimation of $\boldsymbol{\psi}_t$ and v_t over time in $t = 1, ..., T$. We define $D_t = \{D_{t-1}, \boldsymbol{y}_t\}$ and D_0 is the initial prior setting for prior setting of all parameters and $\boldsymbol{\theta}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi}$ from the previous Gibbs steps (if exist, not capable for the first Gibbs step). At each Gibbs step, we calculate the parameters $(\boldsymbol{\psi}_t, v_t)$ conditional on D_t utilizing the forward filtering methodology:

- The posterior distribution for $(\boldsymbol{\psi}_{t-1}, v_{t-1}|D_{t-1})$ is normal/inverse Gamma with marginal distributions as

  – $\boldsymbol{\psi}_{t-1} \sim T_{n_{t-1}}(\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1})$,

  – $v_{t-1}^{-1} \sim Ga(n_{t-1}/2, d_{t-1}/2)$,

  where $\boldsymbol{m}_{t-1}, \boldsymbol{C}_{t-1}$ are the known parameters from D_{t-1}, which is calculated similarly in the previous iteration when calculating $(\boldsymbol{\psi}_{t-1}, v_{t-1})$.

- The prior distribution of $(\boldsymbol{\psi}_t, v_t|D_{t-1})$ is

  – $\boldsymbol{\psi}_t \sim T_{\delta_v n_{t-1}}(\boldsymbol{m}_{t-1}, \boldsymbol{G}_t)$,

  – $v_t^{-1} \sim Ga(\delta_v n_{t-1}/2, \delta_v d_{t-1}/2)$,

  where $\boldsymbol{G}_t = \boldsymbol{C}_{t-1} + \boldsymbol{W}_t$ and $\boldsymbol{W}_t = \frac{1-\delta_W}{\delta_W}\boldsymbol{C}_{t-1}$ as the variance discounting model, so $\boldsymbol{G}_t = \boldsymbol{C}_{t-1}/\delta_W$. δ_W is a known discounting factor set as 0.9 [17]. We define point estimation for v_t as $s_{t-1} = d_{t-1}/n_{t-1}$.

- The distribution of one-step forecast of output $(\hat{\boldsymbol{y}}_t|D_{t-1})$ is $\hat{\boldsymbol{y}}_t \sim T_{\delta_v n_{t-1}}(\boldsymbol{F}_t'\boldsymbol{m}_{t-1}, \boldsymbol{Q}_t)$ where $\boldsymbol{Q}_t = \boldsymbol{F}_t'\boldsymbol{G}_t\boldsymbol{F}_t + s_{t-1}\boldsymbol{R}$, $\boldsymbol{R}$ is the correlation matrix with entries $\boldsymbol{R}_{i,k} = \frac{1}{\sum_{j=1}^q \sigma_j^2} \sum_{j=1}^q \sigma_j^2 \tau_{z_{1j},z_{2j}}^{(j)} R(\boldsymbol{x}_i, \boldsymbol{x}_k|\boldsymbol{\theta}^{(j)})$;

- The estimations of parameters $\boldsymbol{\psi}_t$ and v_t are calculated as follows

$$v_t^{-1} \sim Ga(n_t/2, d_t/2) \tag{4.7}$$

$v_t^{-1} \sim Ga(n_t/2, d_t/2)$ where $n_t = \delta_t n_{t-1} + n$ and $d_t = \delta_v d_{t-1} + s_{t-1}\boldsymbol{e}_t'\boldsymbol{Q}_t'\boldsymbol{e}_t$ where $\boldsymbol{e}_t = \boldsymbol{y}_t - \boldsymbol{F}_t'\boldsymbol{m}_{t-1}$, δ_v is a known discounting factor set as 0.9, and

$$\boldsymbol{\psi}_t \sim T_{n_t}(\boldsymbol{m}_t, \boldsymbol{C}_t) \tag{4.8}$$

where $\boldsymbol{m}_t = \boldsymbol{m}_{t-1} + \boldsymbol{A}_t(\boldsymbol{y}_t - \boldsymbol{F}_t'\boldsymbol{m}_{t-1})$, $\boldsymbol{C}_t = (\boldsymbol{W}_t - \boldsymbol{A}_t\boldsymbol{Q}_t\boldsymbol{A}_t')s_t/s_{t-1}$ with $\boldsymbol{A}_t = \boldsymbol{G}_t\boldsymbol{F}_t\boldsymbol{Q}_t^{-1}$.

After the calculation of forward filtering method, we get the estimations of $\boldsymbol{\psi} = \{\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_T\}$ and $\boldsymbol{v} = \{v_1, ..., v_T\}$. Then we use the backward sampling method to draw samples for $\boldsymbol{\psi} = \{\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_T\}$ and $\boldsymbol{v} = \{v_1, ..., v_T\}$ according to reverse time order as follows:

- t=T, draw v_T^{-1} from $Ga(n_T/1, d_T/2)$, draw $\boldsymbol{\psi}_T|v_T$ from $N(m_T, \boldsymbol{C}_T v_T/s_T)$.
- Set $n_T^* = n_T$, $d_T^* = d_T$, $\boldsymbol{m}_T^* = \boldsymbol{m}_T$, $s_T^* = s_T$. For $t = T-1, ..., 1$, draw samples from the following distribution
  - $v_t^{-1} \sim Ga(n_t^*/2, d_t^*/2)$ where $d_t^* = n_t^* s_t^*$, $n_t^* = (1-\delta_v)n_t + \delta_v n_{t+1}^*$, $\frac{1}{s_t^*} = \frac{1-\delta_v}{s_t} + \frac{\delta_v}{s_{t+1}^*}$;
  - $\psi_t|v_t \sim N(m_t^*, \boldsymbol{C}_t^* v_t/s_t^*)$ where $m_t^* = (1-\delta_w)m_t + \delta_w m_{t+1}^*$, $\boldsymbol{C}_t^* = (1-\delta_w)\frac{\boldsymbol{C}_t}{s_t} + \delta_w^2\frac{\boldsymbol{C}_{t+1}^*}{s_{t+1}^*}$.

After we sample the parameters $(\boldsymbol{\psi}_{(g)}, \boldsymbol{v}_{(g)})$, we use Metropolis-Hasting algorithm to sample $\boldsymbol{\theta}_{(g)}$, $\boldsymbol{\sigma}^{\mathbf{2}}_{(g)}$ and $\boldsymbol{\varphi}_{(g)}$ in sequence. To ease the notation, we denote

$(\boldsymbol{\psi}_{(g-1)}, \boldsymbol{v}_{(g-1)})$, $\boldsymbol{\theta}_{(g-1)}$, $\boldsymbol{\sigma^2}_{(g-1)}$ and $\boldsymbol{\varphi}_{(g-1)}$ as $(\boldsymbol{\psi}, \boldsymbol{v})$, $\boldsymbol{\theta}$, $\boldsymbol{\sigma^2}$ and $\boldsymbol{\varphi}$ in the following Metropolis-Hasting sampling steps. The full conditional distribution of $\boldsymbol{\theta}$ with regard to (4.3) and (4.4) is

$$\begin{aligned}
p(\boldsymbol{\theta}|\boldsymbol{\psi}, \boldsymbol{v}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi}, \boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\psi}, \boldsymbol{v}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi}, \boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}) \\
&\propto \exp(-\frac{1}{2}\sum_{t=p+1}^{T}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{R}^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)/v_t) \\
&\quad \times |\boldsymbol{R}|^{-\frac{T-p}{2}} \times \prod_{t=p+1}^{T}(v_t^{-n/2}) \times \\
&\quad \times \prod_{j=1}^{q}\{\prod_{i=1}^{p}(1-\exp(-\theta_i^{(j)}/4))^{0.9}\exp(-\theta_i^{(j)}/4)\} \\
&\propto |\boldsymbol{R}|^{-\frac{T-p}{2}} \times \exp(-\frac{1}{2}\sum_{t=p+1}^{T}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{R}^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)/v_t) \\
&\quad \times \prod_{j=1}^{q}\{\prod_{i=1}^{p}(1-\exp(-\theta_i^{(j)}/4))^{0.9}\exp(-\theta_i^{(j)}/4)\}
\end{aligned} \tag{4.9}$$

We first set $\eta_i^{(j)} = \log(\theta_i^{(j)})$, $i = 1, ..., p, j = 1, ..., q$ and $\boldsymbol{\eta}^{(j)} = (\eta_1^{(j)}, ..., \eta_p^{(j)})$ [17]. Propose a new value for $\eta_i^{(j)}$ as

$$(\eta_i^{(j)})_* = (\eta_i^{(j)}) + N(0, 0.1^2),$$

then setting $(\theta_i^{(j)})_* = \exp\{(\eta_i^{(j)})_*\}$. Denote $p(\boldsymbol{\theta}^*|\boldsymbol{\psi}, \boldsymbol{v}, \boldsymbol{\sigma^2}, \boldsymbol{\varphi}, \boldsymbol{y})$ as the full conditional distribution of $\boldsymbol{\theta}^*$. With Jacobian

$$J(\boldsymbol{\theta}) = \prod_{j=1}^{q}\prod_{i=1}^{p}\frac{d\eta_i^{(j)}}{d\theta_i^{(j)}} = \prod_{j=1}^{q}\prod_{i=1}^{p}(\theta_i^{(j)})^{-1},$$

the Metropolis-Hasting ratio is

$$r_{\boldsymbol{\theta}} = \mathbf{max}\,\{\frac{p(\boldsymbol{\theta}_*|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\sigma^2},\boldsymbol{\varphi},\boldsymbol{y})J(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\sigma^2},\boldsymbol{\varphi},\boldsymbol{y})J(\boldsymbol{\theta}_*)},1\}.$$

We set $\boldsymbol{\theta}_{(g)} = \boldsymbol{\theta}_*$ with probability $r_{\boldsymbol{\theta}}$, otherwise set $\boldsymbol{\theta}_{(g)} = \boldsymbol{\theta}$.

The full conditional distribution of $\boldsymbol{\sigma^2}$ with regard to (4.3) and (4.20) is

$$\begin{aligned}
p(\boldsymbol{\sigma^2}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{y}) \propto\ & p(\boldsymbol{y}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\sigma^2},\boldsymbol{\varphi},\boldsymbol{\theta}) \times \pi(\boldsymbol{\sigma^2}) \\
\propto\ & \exp(-\frac{1}{2}\sum_{t=p+1}^{T}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{R}^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)/v_t) \\
& \times |\boldsymbol{R}|^{-\frac{T-p}{2}} \times \prod_{t=p+1}^{T}(v_t^{-n/2})\times \\
& \times \prod_{i=1}^{T}(\sigma^2)^{-\alpha_i^{(0)}-1}e^{-\beta_i^{(0)}\sigma_i^2} \\
\propto\ & |\boldsymbol{R}|^{-\frac{T-p}{2}} \times \exp(-\frac{1}{2}\sum_{t=p+1}^{T}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{R}^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)/v_t) \\
& \times \prod_{i=1}^{T}(\sigma^2)^{-\alpha_i^{(0)}-1}e^{-\beta_i^{(0)}\sigma_i^2}
\end{aligned} \tag{4.10}$$

The Metropolis-Hasting ratio is

$$r_{\boldsymbol{\sigma^2}} = \mathbf{max}\,\{\frac{p(\boldsymbol{\sigma^2_*}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{y})}{p(\boldsymbol{\sigma^2}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\theta},\boldsymbol{\varphi},\boldsymbol{y})},1\};$$

The new value can be set as

$$\boldsymbol{\sigma^2_*} = \boldsymbol{\sigma^2} + \boldsymbol{\varepsilon}$$

where each element of $\boldsymbol{\varepsilon}$ is a small random number drawn from normal distribution $N(0, 0.1^2)$. We set $\boldsymbol{\sigma^2_{(g)}} = \boldsymbol{\sigma^2_*}$ with probability $r_{\boldsymbol{\sigma^2}}$, otherwise set $\boldsymbol{\sigma^2_{(g)}} = \boldsymbol{\sigma^2}$.

The full conditional distribution of $\boldsymbol{\varphi}$ with regard to (4.3) and (4.6) is

$$\begin{aligned}
p(\boldsymbol{\varphi}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\sigma^2},\boldsymbol{\theta},\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\sigma^2},\boldsymbol{\varphi},\boldsymbol{\theta}) \times \pi(\boldsymbol{\varphi}) \\
&\propto \exp(-\frac{1}{2}\sum_{t=p+1}^{T}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{R}^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)/v_t) \\
&\quad \times |\boldsymbol{R}|^{-\frac{T-p}{2}} \times \prod_{t=p+1}^{T}(v_t^{-n/2}) \\
&\propto |\boldsymbol{R}|^{-\frac{T-p}{2}} \times \exp(-\frac{1}{2}\sum_{t=p+1}^{T}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)'\boldsymbol{R}^{-1}(\boldsymbol{y}_t - F_t\boldsymbol{\psi}_t)/v_t).
\end{aligned} \tag{4.11}$$

The Metropolis-Hasting ratio is

$$r_{\boldsymbol{\varphi}} = \mathbf{max}\{\frac{p(\boldsymbol{\varphi}_*|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\sigma^2},\boldsymbol{\theta},\boldsymbol{y})}{p(\boldsymbol{\varphi}|\boldsymbol{\psi},\boldsymbol{v},\boldsymbol{\sigma^2},\boldsymbol{\theta},\boldsymbol{y})}, 1\}.$$

where

$$\boldsymbol{\varphi}_* = \boldsymbol{\varphi} + \boldsymbol{\varepsilon}$$

with a range of $\boldsymbol{\varphi}_* \in (0,\pi)$, and each element of $\boldsymbol{\varepsilon}$ is a small random number drawn from normal distribution $N(0, 0.1^2)$. We set $\boldsymbol{\varphi}_{(g)} = \boldsymbol{\varphi}_*$ with probability $r_{\boldsymbol{\varphi}}$, otherwise set $\boldsymbol{\varphi}_{(g)} = \boldsymbol{\varphi}$.

4.1.3 Prediction and Interpolation

Given the estimates for all unknown parameters $\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}}, \hat{\boldsymbol{\varphi}}, \hat{\boldsymbol{\psi}}, \hat{\boldsymbol{v}}$ in the model, for a new input $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z})$, its prediction at time t conditional on training data set $(\boldsymbol{w}_1, ..., \boldsymbol{w}_n)$

and all previous time series values $y(\boldsymbol{w}, 1), ..., y(\boldsymbol{w}, t-1)$ is

$$\hat{y}(\boldsymbol{w}, t) = \sum_{j=1}^{h} \hat{\psi}_{t,j} y(\boldsymbol{w}, t-j) + \boldsymbol{r}'_{\boldsymbol{w}} \hat{\boldsymbol{R}}^{-1} \hat{\boldsymbol{\varepsilon}}_t \tag{4.12}$$

and variance

$$\sigma_t^2(\boldsymbol{w}) = \hat{v}_t(1 - \boldsymbol{r}'_{\boldsymbol{w}} \hat{\boldsymbol{R}}^{-1} \boldsymbol{r}_{\boldsymbol{w}})$$

where

$$\hat{\boldsymbol{R}} = \boldsymbol{R}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\sigma}}^{\boldsymbol{2}}, \hat{\boldsymbol{\varphi}}),$$

$$\boldsymbol{r}_{\boldsymbol{w}} = (r(y(\boldsymbol{w}, \boldsymbol{w}_1)), ..., r(y(\boldsymbol{w}, \boldsymbol{w}_n)))',$$

$$\hat{\boldsymbol{\varepsilon}}_t = (\hat{\varepsilon}_t(\boldsymbol{w}_1), ..., \hat{\varepsilon}_t(\boldsymbol{w}_n))',$$

$$r(y(\boldsymbol{w}, \boldsymbol{w}_i)) = \frac{1}{\sum_{j=1}^{q} \hat{\sigma}_j^2} \sum_{j=1}^{q} \hat{\sigma}_j^2 \hat{\tau}^{(j)}_{z_{ij}, z_j} R(\boldsymbol{x}_i, \boldsymbol{x} | \hat{\boldsymbol{\theta}}^{(j)}),$$

$$\hat{\varepsilon}_t(\boldsymbol{w}_i) = y(\boldsymbol{w}_i, t) - \sum_{j=1}^{h} \hat{\psi}_{t,j} y(\boldsymbol{w}_i, t-j), 1 \le i \le n.$$

For interpolation, we suppose that we make the prediction at $\boldsymbol{w} = \boldsymbol{w}_i$, $1 \le i \le n$, then we have

$$\boldsymbol{r}'_{\boldsymbol{w}} \hat{\boldsymbol{R}}^{-1} = (\hat{\boldsymbol{R}}^{-1} \boldsymbol{r}_{\boldsymbol{w}})' = (\hat{\boldsymbol{R}}^{-1} \hat{\boldsymbol{R}}_i)' = \boldsymbol{e}'_i$$

where $\hat{\boldsymbol{R}}_i$ is the i-th column of $\hat{\boldsymbol{R}}$ and $\boldsymbol{e}_i$ is the $n \times 1$ vector with the i-th entry as 1 and other entries as 0. Equation (4.12) can be written as

$$\begin{aligned}
\hat{y}(\boldsymbol{w}_i, t) &= \sum_{j=1}^{h} \hat{\psi}_{t,j} y(\boldsymbol{w}, t-j) + \hat{\varepsilon}_t(\boldsymbol{w}_i) \\
&= \sum_{j=1}^{h} \hat{\psi}_{t,j} y(\boldsymbol{w}, t-j) + (y(\boldsymbol{w}_i, t) - \sum_{j=1}^{h} \hat{\psi}_{t,j} y(\boldsymbol{w}_i, t-j)) \\
&= y(\boldsymbol{w}_i, t).
\end{aligned}$$

Thus, the prediction at $\boldsymbol{w} = \boldsymbol{w}_i$, $1 \le i \le n$ is the observed value $y(\boldsymbol{w}_i, t)$, which interpolates the data.

4.1.4 Choice of Hyper-parameters in Prior Distributions

In this section, we test the performance of the hyper-parameters in all the prior distributions we have discussed before, and give a suggestion how to choose the hyper-parameters. Referring to [17], we use the initial settings as $Beta(2, 1.9)$ for $\rho_i^{(j)}$, $\rho_i^{(j)} = \exp(-\theta_i^{(j)}/4)$, inverse Gamma for $\boldsymbol{\sigma}^2$, i.e. $\sigma_i^2 \sim IG(10, 0.1)$, normal/inverse Gamma for $(\boldsymbol{v}, \boldsymbol{\psi})$, i.e. $v_0^{-1} \sim Ga(1/2, 1/2)$ and $\boldsymbol{\psi} \sim N((1\ 1)', diag(10, 10))$. These prior hyper-parameters are chosen according to [17] and [11]. To test if these hyper-parameters are suitable for the model, we change one hyper-parameter and remain others as the initial setting and calculate the corresponding root mean square prediction error on each simulator over 50 replications.

In this section, we test different priors for $\boldsymbol{\theta}$, $\boldsymbol{\sigma}^2$, $\boldsymbol{v}$ and $\boldsymbol{\psi}$ with the following two simulators. The first simulator ([3], [26]) is

$$\begin{aligned} y(\boldsymbol{w}, t) =& 1.5g_1(x_1 t) + 3g_2(x_2 + z_3) + 1.2g_4(x_3) + \\ & 2g_2(x_4 z_2) + 4g_2(x_7)t + 1.4g_2(z_4 + z_5) + g_2(z_6 + 1.6z_7) + \\ & 0.2g_3(x_5 + x_6) + g_1(x_8)g_2(x_9) \end{aligned} \tag{4.13}$$

where

$$\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z}), x_i \in (0, 1), i = 1, ..., 9, z_j \in \{0.1, 0.4, 0.9\}, j = 1, ..., 7,$$

g_1, g_2, g_3, g_4 are defined as

$$\begin{aligned} & g_1(x) = x; \\ & g_2(x) = (2x - 1)^2; \\ & g_3(x) = sin(2\pi x)/[2 - sin(2\pi x)]; \\ & g_4(x) = 0.1sin(2\pi x) + 0.2cos(2\pi x) + 0.3[sin(2\pi x)]^2 + 0.4[cos(2\pi x)]^3 + 0.5[sin(2\pi x)]^3, \end{aligned}$$

and time points are 200 equally spaced points in the range $t \in (0, 1)$. In each replication, for the training data of size 81, the designs for quantitative factors are maximin Latin hypercube designs while the design for qualitative factors is 81-run three-level fractional factorial design [15]. For the test data of size 1620, the designs for quantitative factors are random Latin hypercube designs while the qualitative factors are randomly generated by the MATLAB function $'randi'$.

The second simulator [20] is

$$y(\boldsymbol{w}, t) = m(x_1, x_2) + m(x_1, t) + m(x_4, z_4) + m(z_1, t) + m(x_5 - x_2, z_2) + m(x_3, z_3) \tag{4.14}$$

where

$$\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z}), \boldsymbol{x} \in (0,2) \times (0,4) \times (0,1) \times (0,1) \times (0,1),$$

$$z_i \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5\}, i = \{1, ..., 4\},$$

$m(\cdot, \cdot)$ is the Mccormick function defined as

$$m(x_1, x_2) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1.$$

and time points are 200 equally spaced points in the range $t \in [1.5, 3.5]$. In each replication, for the training data of size 216, the designs for quantitative factors are maximin Latin hypercube designs while the design for qualitative factors is 216-run six-level factional factorial design [15]. For the test data of size 1080, the designs for quantitative factors are random Latin hypercube designs while the qualitative factors are randomly generated by the MATLAB function $'randi'$.

As in Section 4.2.1, we assume Beta distribution $Beta(a_0, b_0)$ for $\rho_i^{(j)}$, the range parameter, where $\rho_i^{(j)} = \exp(-\theta_i^{(j)}/4)$. Thus, the joint prior distribution of $\boldsymbol{\theta}$ is

$$p(\boldsymbol{\theta}) \propto \prod_{j=1}^{q} \prod_{i=1}^{p} (1 - \exp(-\theta_i^{(j)}/4))^{b_0 - 1} \exp(-\theta_i^{(j)}/4)^{a_0 - 1}.$$

With the reference of [17] and [11], we choose several pairs of hyper-parameters for the prior of $\boldsymbol{\theta}$ as Table 4.1. From Figure 4.1 and Figure 4.2, we recommend the choice

a_0	0.5	0.5	1	2	2	2	3	5
b_0	0.5	2	3	0.5	1.9	5	1	2

Table 4.1: The values of the tested hyper-parameters a_0 and b_0 in the prior of θ in the $DLM - ADGP$ model.

of $a_0 = 2, b_0 = 1.9$ which shows the best performance in the comparison.

As in Section 4.2.1, we assume inverse gamma distribution for $\boldsymbol{\sigma}^2$, the covariance component of qualitative factors,

$$\sigma_i^2 \sim IG(\alpha_i^{(0)}, \beta_i^{(0)}), 1 \leq i \leq q.$$

To test the robustness of $\boldsymbol{\sigma}^2$, we choose several pairs of hyper-parameters for the prior with the reference of [10], [17] and [11] as Table 4.2. From Figure 4.3 and

$\alpha_i^{(0)}$	0.1	0.1	0.1	1	1	1	10	10	10
$\beta_i^{(0)}$	0.1	1	10	0.1	1	10	0.1	1	10

Table 4.2: The values of the tested hyper-parameters $\alpha_i^{(0)}$ and $\beta_i^{(0)}$ in the prior of $\boldsymbol{\sigma}^2$ in the $DLM - ADGP$ model.

Figure 4.4, we can see that in simulator (4.13), the choice of $\alpha_i^{(0)} = 10, \beta_i^{(0)} = 0.1$ shows the best performance in the comparison, while in simulator (4.14), the choice of $\alpha_i^{(0)} = 10, \beta_i^{(0)} = 0.1$ is only slightly better than other choices.

As in Section 4.2.1, we assume normal/inverse Gamma for $(\boldsymbol{\psi}, \boldsymbol{v})$ with prior marginal distribution of $\boldsymbol{v}$ as $v_0^{-1} \sim Ga(n_0/2, d_0/2)$, where $\boldsymbol{v}$ is the covariance component vector, $\boldsymbol{\psi}$ is the initial mean vector. We choose the hyper-parameters of the prior distributions according to [17] as Table 4.3. In Figure 4.5 and Figure 4.6, we

n_0	0.1	0.1	0.1	1	1	1	10	10	10
d_0	0.1	1	10	0.1	1	10	0.1	1	10

Table 4.3: The values of the tested hyper-parameters n_0 and d_0 in the prior of $\boldsymbol{v}^{-1}$ in the $DLM - ADGP$ model.

can see that small n_0 is a better choice. We recommend the $v_0^{-1} \sim Ga(0.05, 5)$.

As in Section 4.2.1, $\boldsymbol{\psi}$ follows a 2-dimensional normal distribution. We choose the hyper-parameters of the prior distributions according to [17] as Table 4.4. According

	Group1	Group2	Group3	Group4	Group5	Group6
m_0	1	1	1	5	5	5
$\boldsymbol{C}_0$	diag(1,1)	diag(5,5)	diag(10,10)	diag(1,1)	diag(5,5)	diag(10,10)

Table 4.4: The values of the tested hyper-parameters m_0 and $\boldsymbol{C}_0$ in the prior of $\boldsymbol{\psi}$ in the $DLM - ADGP$ model.

to the Figure 4.7 and Figure 4.8, we recommend the choice of $m_0 = (1,1)', \boldsymbol{C}_0 = diag(10,10)$ which shows the best performance in the comparison.

In conclusion, the choice of hyper-parameters will affect the model and should be chosen carefully according to the data. According to the tests we have done, we recommend $Beta(2,1.9)$ for the range parameter $\rho_i^{(j)}$, $IG(10,0.1)$ for the variance component $\boldsymbol{\sigma}^2$, $Ga(0.05,5)$ for prior marginal distribution of $\boldsymbol{v}$, $N((1,1)', diag(10,10))$ for $\boldsymbol{\psi}$.

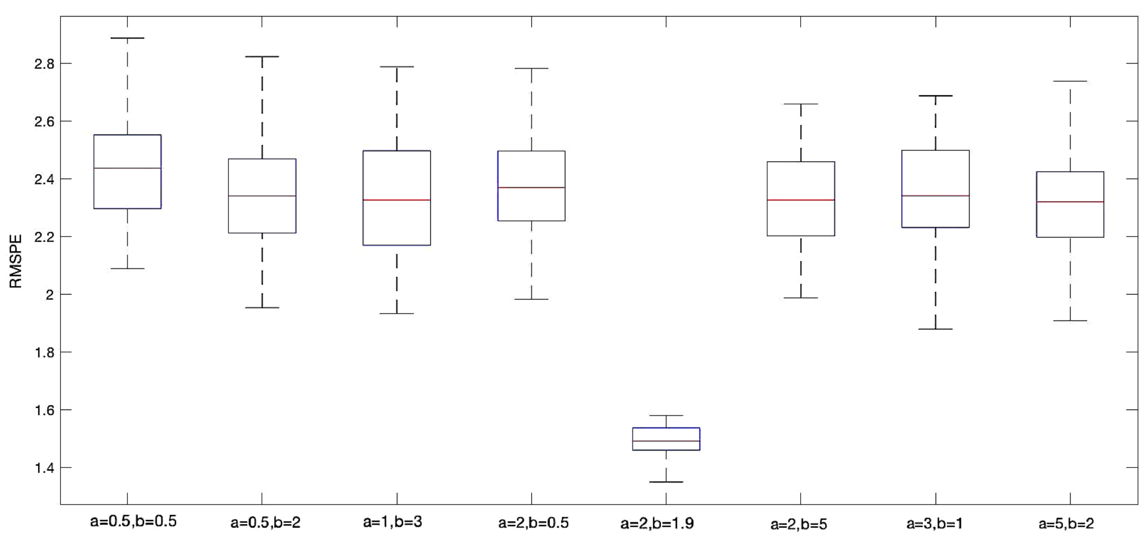

Figure 4.1: The boxplot of the RMSPEs computed from the simulator (4.13) with different hyper-parameters for $\boldsymbol{\theta}$ over 50 replications.

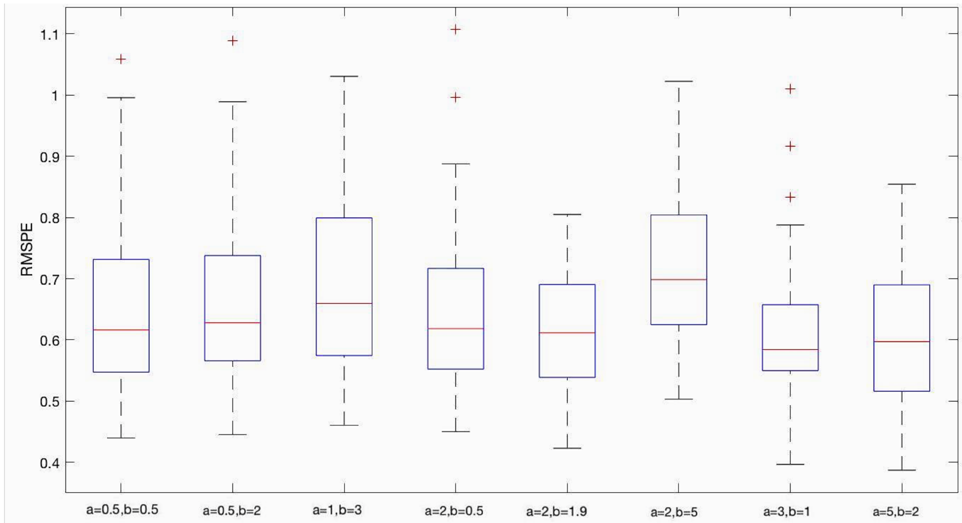

Figure 4.2: The boxplot of the RMSPEs computed from the simulator (4.14) with different hyper-parameters for $\boldsymbol{\theta}$ over 50 replications.

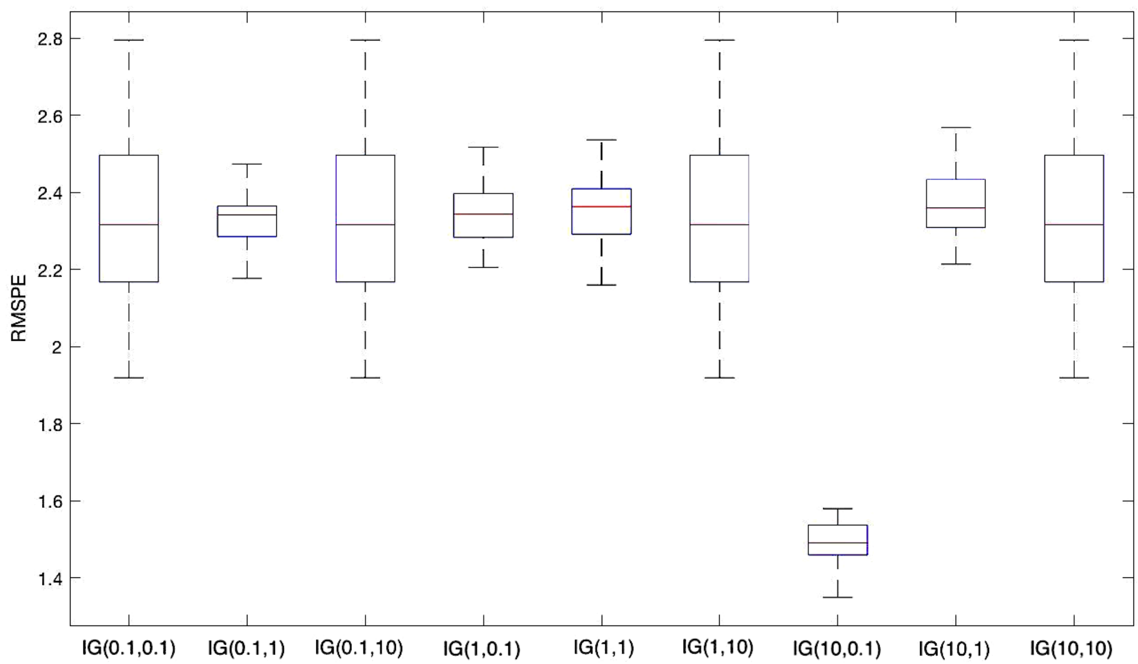

Figure 4.3: The boxplot of the RMSPEs computed from the simulator (4.13) with different hyper-parameters for $\boldsymbol{\sigma}^2$ over 50 replications.

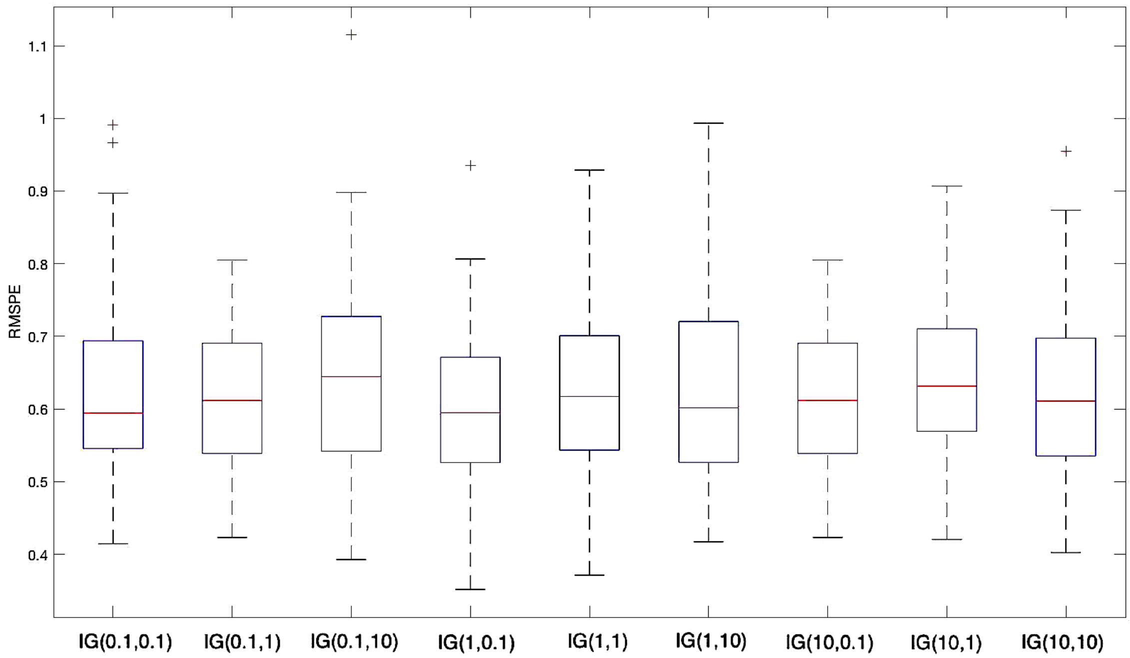

Figure 4.4: The boxplot of the RMSPEs computed from the simulator (4.14) with different hyper-parameters for $\boldsymbol{\sigma}^2$ over 50 replications.

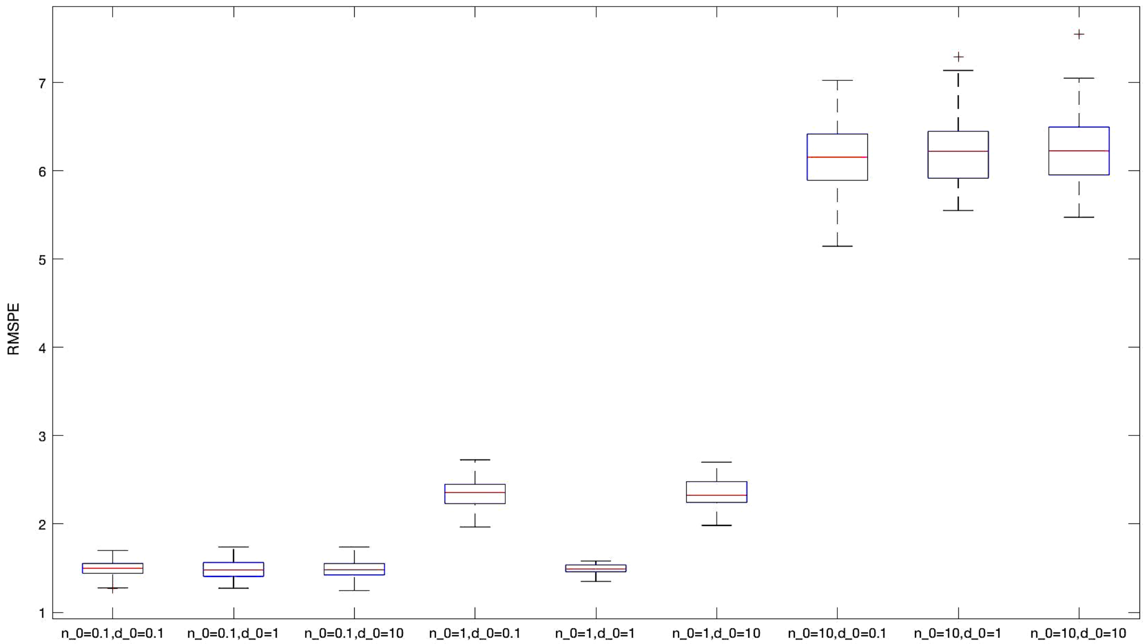

Figure 4.5: The boxplot of the RMSPEs computed from the simulator (4.13) with different hyper-parameters for $\boldsymbol{v}$ over 50 replications.

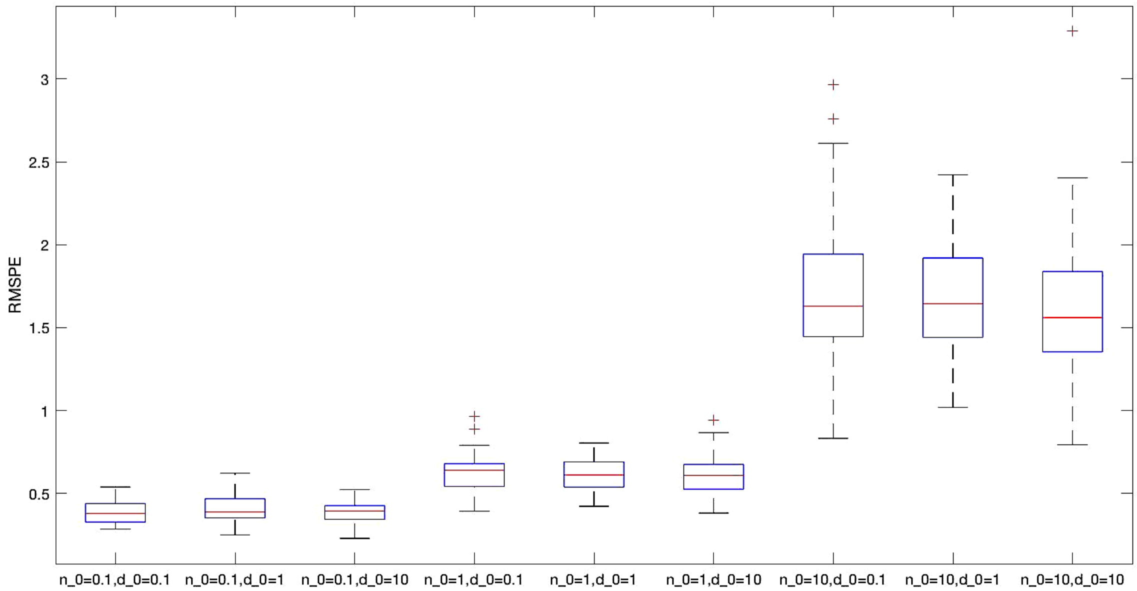

Figure 4.6: The boxplot of the RMSPEs computed from the simulator (4.14) with different hyper-parameters for $\boldsymbol{v}$ over 50 replications.

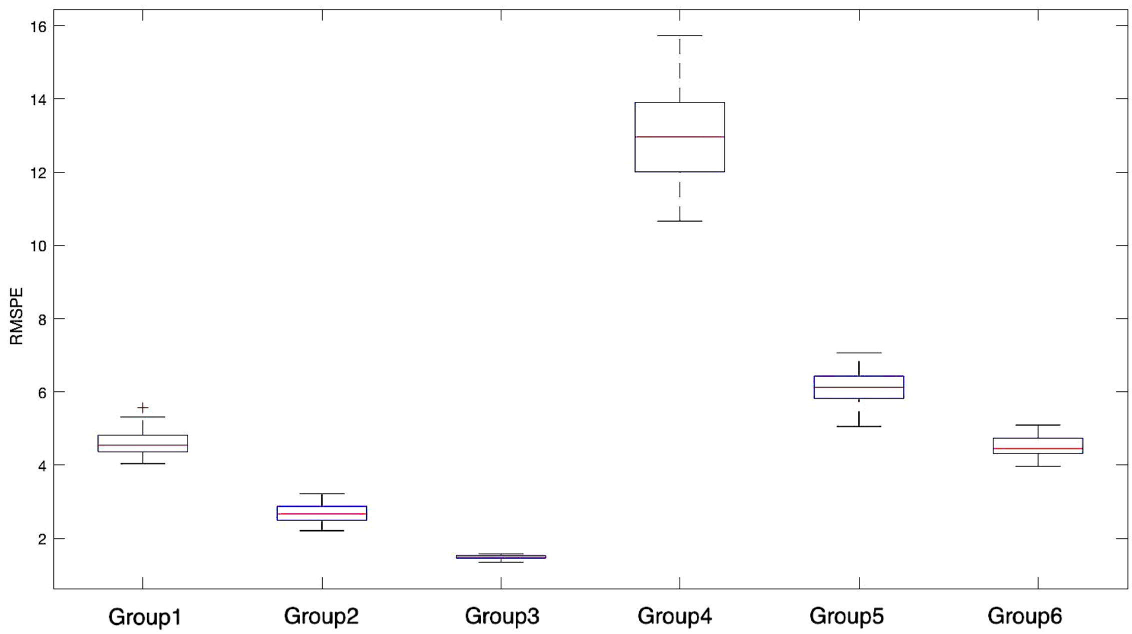

Figure 4.7: The boxplot of the RMSPEs computed from the simulator (4.13) with different hyper-parameters for $\boldsymbol{\psi}$ over 50 replications.

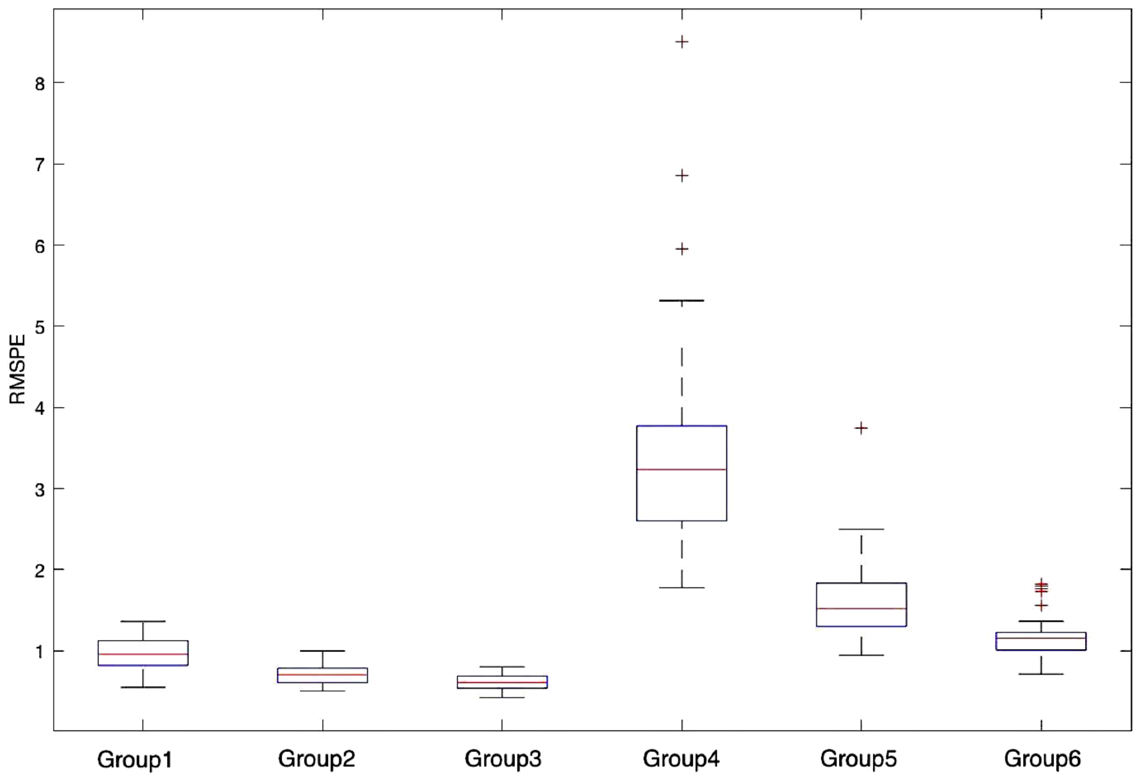

Figure 4.8: The boxplot of the RMSPEs computed from the simulator (4.14) with different hyper-parameters for $\boldsymbol{\psi}$ over 50 replications.

4.2 Singular Value Decomposition Based Additive Gaussian Process Model

4.2.1 Model Description

We consider an input with p quantitative factors $\boldsymbol{x} = (x_1, ..., x_p)$ and q qualitative factors $\boldsymbol{z} = (z_1, ..., z_q)$ with $m_1, ..., m_q$ levels respectively, and define the input as $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z})$. Consider a simulator built by n training points $\boldsymbol{W} = (\boldsymbol{w}_1, ..., \boldsymbol{w}_n)$ and the corresponding $T \times n$ time-series response matrix $\boldsymbol{Y} = (\boldsymbol{y}_1, ..., \boldsymbol{y}_n)$.

The SVD on $\boldsymbol{Y}$ is

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}'$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, ..., \boldsymbol{u}_k]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_k]$ are $T \times k$ and $n \times k$ column orthogonal matrix respectively, $k = \mathbf{min}\,(n, T)$, and $\boldsymbol{D} = diag(d_1, ..., d_k)$ is the diagonal matrix of singular values in decreasing order. We define $\boldsymbol{B} = [\boldsymbol{b}_1, ..., \boldsymbol{b}_l] = \boldsymbol{U}^*\boldsymbol{D}^*$, where $\boldsymbol{U}^* = [\boldsymbol{u}_1, ..., \boldsymbol{u}_l]$ and $\boldsymbol{D}^* = diag(d_1, ..., d_l)$ are the first l vectors of $\boldsymbol{U}, \boldsymbol{D}$, and we also define $\boldsymbol{V}^* = [\boldsymbol{v}_1, ..., \boldsymbol{v}_l]$. We choose the number of significant singular values l by the cumulative percentage as following

$$l = min\{l : \frac{\sum_{i=1}^{l} d_i}{\sum_{i=1}^{k} d_i} > \gamma\}$$

where γ is a pre-specified threshold of explained variation. The suggested values are between 0.9 and 0.99 [11]. We use $\gamma = 0.9$.

To represent the T-dimensional output $\boldsymbol{y}(\boldsymbol{w})$ at input $\boldsymbol{w}$, we have

$$\boldsymbol{y}(\boldsymbol{w}) = \sum_{i=1}^{l} \boldsymbol{b}_i c_i(\boldsymbol{w}) + \boldsymbol{\varepsilon} \tag{4.15}$$

where $\boldsymbol{\varepsilon}$ is the residual,

$$\boldsymbol{\varepsilon} \sim N(0, \lambda_{\varepsilon}^{-1}\boldsymbol{I}).$$

We assume that $c_i(\cdot)$ follows a zero mean GP for $1 \leq i \leq l$,

$$c_i(\cdot) \sim N(0, \lambda_i^{-1} K_i) \tag{4.16}$$

with the correlation function K_i defined as

$$K(\boldsymbol{w}_j, \boldsymbol{w}_h; \boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}) = \sum_{s=1}^{q} \frac{(\sigma_s^2)^{(i)}}{\sum_{s=1}^{q} (\sigma_s^2)^{(i)}} \tau_{z_{js}, z_{hs}}^{(i)} \exp(-\sum_{k=1}^{p} \theta_{s,k}^{(i)} |x_{jk} - x_{hk}|^2),$$

where $[\boldsymbol{T}_s^{(i)}]_{j,k} = \tau_{z_{js}, z_{ks}}^{(i)}$, $i = 1, ..., l$, $s = 1, ..., q$, are the qualitative correlation parameters of additive GP model using hypersphere decomposition, $(\sigma_s^2)^{(i)}, 1 \leq s \leq q$ as the variance component for qualitative factors, and $\theta_{s,k}^{(i)}$, $i = 1, ..., l$, $s = 1, ..., q$, are the roughness parameters. Denote $(\boldsymbol{\sigma}^2)^{(i)} = ((\sigma_1^2)^{(i)}, ..., (\sigma_q^2)^{(i)})$ and $\boldsymbol{\theta}^{(i)} = (\theta_{s,k}^{(i)}, 1 \leq s \leq q, 1 \leq k \leq p)$ for $1 \leq i \leq l$. For the $n \times n$ training data $\boldsymbol{W}$, we define the $n \times n$ correlation matrix K_i, $1 \leq i \leq l$ as

$$[K_i]_{j,h} = K(\boldsymbol{w}_j, \boldsymbol{w}_h; \boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}).$$

According to [22], we treat $\boldsymbol{v}_i$, the i-th column of $\boldsymbol{V}$, as the realization of GP $c_i(\cdot)$ at $\boldsymbol{W}$, i.e. $\boldsymbol{v}_i = (c_i(\boldsymbol{w}_1), ..., c_i(\boldsymbol{w}_n))'$.

4.2.2 Parameter Estimation

According to (4.16), we have

$$\pi(\boldsymbol{v}_i|\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}, \lambda_i) \propto |K_i|^{-1/2}\lambda_i^{\frac{n}{2}} \exp\{-\frac{\psi_i}{2}\lambda_i\} \tag{4.17}$$

where $\psi_i = \boldsymbol{v}_i' K_i^{-1}\boldsymbol{v}_i$. Note that $(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)})$ are embedded in K_i. Also we assume that the unknown parameters are mutually independent, so we have

$$\pi(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}, \lambda_i) = \pi(\boldsymbol{\theta}^{(i)})\pi(\boldsymbol{T}^{(i)})\pi((\boldsymbol{\sigma^2})^{(i)})\pi(\lambda_i).$$

Assume the commonly used Gamma prior for λ_i, i.e. $\lambda_i \sim Ga(a_i, b_i)$,

$$\pi(\lambda_i) \propto (\lambda_i)^{a_i-1} \exp\{-b_i\lambda_i\},$$

we calculate the marginal posterior distribution of $(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)})$ through the integration over λ_i. The Gamma function is

$$\Gamma(a) = \int_0^\infty x^{a-1} \exp(-x)dx, a > 0,$$

for $b > 0$, we have the following equation after some simple calculation,

$$\int_0^\infty x^{a-1} \exp(-bx)dx = b^{-a}\Gamma(a). \tag{4.18}$$

Given (4.18), we have

$$\begin{aligned}
&\pi(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}|\boldsymbol{v}_i) \\
&\propto \int_0^\infty \pi(\boldsymbol{v}_i|\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}, \lambda_i)\pi(\boldsymbol{\theta}^{(i)})\pi(\boldsymbol{T}^{(i)})\pi((\boldsymbol{\sigma^2})^{(i)})\pi(\lambda_i)d(\lambda_i) \\
&\propto \int_0^\infty |K_i|^{-1/2}(\lambda_i)^{\frac{n}{2}+a_i-1}\exp\{-\frac{\psi_i+2b_i}{2}\lambda_i\}\pi(\boldsymbol{\theta}^{(i)})\pi(\boldsymbol{T}^{(i)})\pi((\boldsymbol{\sigma^2})^{(i)})d(\lambda_i) \\
&\propto |K_i|^{-1/2}\pi(\boldsymbol{\theta}^{(i)})\pi(\boldsymbol{T}^{(i)})\pi((\boldsymbol{\sigma^2})^{(i)})\int_0^\infty (\lambda_i)^{\frac{n}{2}+a_i-1}\exp\{-\frac{\psi_i+2b_i}{2}\lambda_i\}d(\lambda_i) \\
&\propto |K_i|^{-1/2}(\frac{\psi_i+2b_i}{2})^{-(\frac{n}{2}+a_i)}\Gamma(\frac{n}{2}+a_i)\pi(\boldsymbol{\theta}^{(i)})\pi(\boldsymbol{T}^{(i)})\pi((\boldsymbol{\sigma^2})^{(i)}) \\
&\propto |K_i|^{-1/2}(\frac{\psi_i+2b_i}{2})^{-(\frac{n}{2}+a_i)}\pi(\boldsymbol{\theta}^{(i)})\pi(\boldsymbol{T}^{(i)})\pi((\boldsymbol{\sigma^2})^{(i)}).
\end{aligned}$$

We specify the prior of λ_i as $Ga(a_i = 0, b_i = 0)$ according to [1]. Thus, we have $\pi(\lambda_i) \propto \lambda_i^{-1}$ and the marginal posterior distribution of $(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)})$ as

$$\pi(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}|\boldsymbol{v}_i) \propto |K_i|^{-1/2}(\frac{\psi_i}{2})^{-\frac{n}{2}}\pi(\boldsymbol{\theta}^{(i)})\pi(\boldsymbol{T}^{(i)})\pi((\boldsymbol{\sigma^2})^{(i)}). \tag{4.19}$$

Assume the prior for the inverse of each element $\theta_{s,k}^{(i)}$ in unknown parameter $\boldsymbol{\theta}^{(i)}$ is independent Gamma distribution, i.e. $\pi(1/\theta_{s,k}^{(i)}) \sim Ga(\alpha_{\theta,s,k}^{(i)}, \beta_{\theta,s,k}^{(i)})$. For the elements in parameter $(\boldsymbol{\sigma}^2)^{(i)}$, we choose the commonly used independent inverse gamma distributions, that is

$$(\sigma_s^2)^{(i)} \sim IG(\alpha_s^{(i)}, \beta_s^{(i)}).$$

The prior joint distribution of $(\boldsymbol{\sigma}^2)^{(i)}$ is

$$\pi((\boldsymbol{\sigma}^2)^{(i)}) \propto \prod_{s=1}^{q}((\sigma_s^2)^{(i)})^{-\alpha_s^{(i)}-1}\exp\{-\beta_s^{(i)}(\sigma_s^2)^{(i)}\}. \tag{4.20}$$

We use the hypersphere decomposition to build the correlation matrix $\boldsymbol{T}_j^{(i)}$ for the

j-th qualitative factor as in (2.24) [26]. Thus, $\boldsymbol{T}^{(i)} = [\boldsymbol{T}_j^{(i)}, 1 \le j \le q]$ is controlled by parameters $\varphi_{j,r,s}^{(i)} \in (0, \pi), 1 \le s \le r, 1 \le r \le m_j$. We treat the parameters in $\boldsymbol{T}_j^{(i)}$ as $\boldsymbol{\varphi}_j^{(i)} = \{\boldsymbol{\varphi}_{j,2}^{(i)}, ..., \boldsymbol{\varphi}_{j,m_i}^{(i)}\}$, $\boldsymbol{\varphi}_{j,r}^{(i)} = (\varphi_{j,r,1}^{(i)}, ..., \varphi_{j,r,r-1}^{(i)})'$ for $2 \le r \le m_i$. All elements in $\boldsymbol{\varphi}^{(i)}$ are mutually independent. We assume the prior distribution of the element in $\boldsymbol{\varphi}^{(i)}$ to be $\varphi_{j,r,s}^{(i)} \sim Unif(0, \pi)$, and $p(\varphi_{j,r,s}^{(i)}) = \frac{1}{\pi}$, so the prior distribution of $\boldsymbol{\varphi}_j^{(i)}$ can be written as

$$\pi(\boldsymbol{\varphi}_j^{(i)}) = \prod_{r=2}^{m_j} \prod_{s=1}^{r-1} \pi(\varphi_{j,r,s}^{(i)}) = (\frac{1}{\pi})^{m_j(m_j-1)/2}. \tag{4.21}$$

Thus, we have the prior distribution of $\boldsymbol{T}^{(i)}$ as

$$\pi(\boldsymbol{T}^{(i)}) = \pi(\boldsymbol{\varphi}^{(i)}) \propto 1.$$

With the prior distributions and (4.19), we can get the MAP estimators of $(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)})$ as

$$(\hat{\boldsymbol{\theta}}^{(i)}, \hat{\boldsymbol{T}}^{(i)}, (\hat{\boldsymbol{\sigma}}^2)^{(i)}) = \underset{\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}}{\text{argmax}} \log(\pi(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)} | \boldsymbol{v}_i)), \tag{4.22}$$

where

$$\log(\pi(\boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)} | \boldsymbol{v}_i)) \propto -\frac{1}{2}|K_i| - \frac{n}{2}\log(\frac{\psi_i}{2}) + \log(\pi(\boldsymbol{\theta}^{(i)})) + \log(\pi((\boldsymbol{\sigma^2})^{(i)}))).$$

The posterior distribution of λ_i according to equation (4.17) with plug-in value of

$(\hat{\boldsymbol{\theta}}^{(i)}, \hat{\boldsymbol{T}}^{(i)}, (\hat{\boldsymbol{\sigma}}^2)^{(i)})$ in (4.22) is

$$\begin{aligned}\pi(\lambda_i|\boldsymbol{v}_i, \hat{\boldsymbol{\theta}}^{(i)}, \hat{\boldsymbol{T}}^{(i)}, (\hat{\boldsymbol{\sigma}}^2)^{(i)}) &\propto \pi(\boldsymbol{v}_i|\lambda_i, \hat{\boldsymbol{\theta}}^{(i)}, \hat{\boldsymbol{T}}^{(i)}, (\hat{\boldsymbol{\sigma}}^2)^{(i)}) \times \pi(\lambda_i) \\ &\propto |K_i|^{-1/2} \lambda_i^{\frac{n}{2}} \exp\{-\frac{\psi_i}{2}\lambda_i\}\lambda_i^{-1} \\ &\propto \lambda_i^{\frac{n}{2}-1} \exp\{-\frac{\psi_i}{2}\lambda_i\}.\end{aligned}$$

Thus, $\lambda_i \sim Ga(n/2, \psi_i/2)$, we can get $\hat{\lambda}_i = \frac{n}{\psi_i}$.

The predictive distribution of any new input $\boldsymbol{w}_0$ follows a student-t distribution [22] with mean

$$\hat{c}_i(\boldsymbol{w}_0) = \boldsymbol{k}_i'(\boldsymbol{w}_0)\boldsymbol{K}_i^{-1}\boldsymbol{v}_i,$$

and variance

$$\hat{\sigma}_i^2(\boldsymbol{w}_0) = \frac{\psi_i(1 - \boldsymbol{k}_i'(\boldsymbol{w}_0)\boldsymbol{K}_i^{-1}\boldsymbol{k}_i(\boldsymbol{w}_0))}{n}$$

where $\hat{c}_i(\boldsymbol{w}_0) = \boldsymbol{k}_i'(\boldsymbol{w}_0)\boldsymbol{K}_i^{-1}\boldsymbol{v}_i$ and
$\boldsymbol{k}_i(\boldsymbol{w}_0) = [K(\boldsymbol{w}_0, \boldsymbol{w}_1; \boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)}), ..., K(\boldsymbol{w}_0, \boldsymbol{w}_n; \boldsymbol{\theta}^{(i)}, \boldsymbol{T}^{(i)}, (\boldsymbol{\sigma^2})^{(i)})]'$.

We define the vectorization of response matrix as $\boldsymbol{y}^n = (\boldsymbol{y}_1', ..., \boldsymbol{y}_n')'$, thus we have

$$\boldsymbol{y}^n \sim N((\boldsymbol{I}_n \otimes \boldsymbol{B})vec((\boldsymbol{V}^*)'), \lambda_\varepsilon^{-1}\boldsymbol{I}_{nT}).$$

We define $\boldsymbol{r} = \boldsymbol{y}^n - (\boldsymbol{I}_n \otimes \boldsymbol{B})vec((\boldsymbol{V}^*)')$ as the vectorization of the residual matrix $\boldsymbol{Y} - \boldsymbol{B}(\boldsymbol{V}^*)'$. The notation $\otimes$ represents the Kronecker product and $vec(\cdot)$ represents the vectorization of a matrix. We assume the Gamma distribution prior $\lambda_\varepsilon \sim Ga(a, b)$, the posterior distribution of the precision λ_ε is

$$\begin{aligned}\pi(\lambda_\varepsilon|\boldsymbol{Y}) &\propto \pi(\boldsymbol{Y}|\lambda_\varepsilon)\pi(\lambda_\varepsilon)\\ &\propto \lambda_\varepsilon^{\frac{nT}{2}} \exp(-\frac{\lambda_\varepsilon \boldsymbol{r}'\boldsymbol{r}}{2})\lambda_\varepsilon^{a-1}\exp(-b\lambda_\varepsilon)\\ &\propto \lambda_\varepsilon^{\frac{nT}{2}+a-1}\exp(-\frac{(\boldsymbol{r}'\boldsymbol{r}+2b)\lambda_\varepsilon}{2})\end{aligned} \tag{4.23}$$

Thus, $\lambda_\varepsilon \sim Ga(\frac{nT}{2}+a, \frac{\boldsymbol{r}'\boldsymbol{r}}{2}+b)$. We can get the estimate of λ_ε as

$$\hat{\lambda}_\varepsilon = \frac{nT+2a}{\boldsymbol{r}'\boldsymbol{r}+2b}.$$

4.2.3 Prediction

Use the result of Lemma 2.2.1 in (4.15), we can obtain the predicted response at any new input $\boldsymbol{w}_0$ following multivariate normal with mean

$$\hat{\boldsymbol{\mu}}(\boldsymbol{w}_0) = \boldsymbol{B}\hat{\boldsymbol{c}}(\boldsymbol{w}_0|\boldsymbol{C}, \hat{\boldsymbol{\rho}}, \hat{\lambda}_\varepsilon) \tag{4.24}$$

and variance

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{B}diag(\hat{\sigma}_1^2(\boldsymbol{w}_0), ..., \hat{\sigma}_l^2(\boldsymbol{w}_0))\boldsymbol{B}' + \hat{\lambda}_\varepsilon^{-1}\boldsymbol{I}_T,$$

where $\hat{\boldsymbol{c}}(\boldsymbol{w}_0|\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{\rho}) = [\hat{c}_1(\boldsymbol{w}_0), ..., \hat{c}_l(\boldsymbol{w}_0)]'$ and $\hat{\boldsymbol{\rho}} = \{\hat{\boldsymbol{\theta}}^{(i)}, \hat{\boldsymbol{T}}^{(i)}, (\hat{\boldsymbol{\sigma}}^2)^{(i)}, \hat{\lambda}_i, 1 \le i \le l\}$.

4.2.4 Choice of Hyper-parameters in Prior Distributions

When considering the point estimation of the response at a new input as shown in (4.24) , we need to choose the hyper-parameters embedded in the prior distributions of $\boldsymbol{\theta}^{(i)}$ and $(\boldsymbol{\sigma}^2)^{(i)}$, $1 \le i \le l$. Recall from Section 4.3.2, we have each element of

$(\boldsymbol{\sigma}^2)^{(i)}$ follows inverse Gamma distribution $(\sigma_s^2)^{(i)} \sim IG(\alpha_s^{(i)}, \beta_s^{(i)})$ and each element of $\boldsymbol{\theta}^{(i)}$ follows Gamma distribution $\pi(1/\theta_{s,k}^{(i)}) \sim Ga(\alpha_{\theta,s,k}^{(i)}, \beta_{\theta,s,k}^{(i)})$. When we test the effectiveness of the choices for the prior of one hyper-parameter, we set flat prior for the other, i.e. we set $\pi(\boldsymbol{\theta}^{(i)}) \propto 1$ when we test which of the priors of $(\boldsymbol{\sigma}^2)^{(i)}$ gives a better performance. Considering the mean and variance of the prior distributions, we choose the hyper-parameters as Table 4.5 and Table 4.6 referring to [10], [11], [14] and [17].

$\alpha_s^{(i)}$	0.1	0.1	0.1	1	1	1	10	10	10	3
$\beta_s^{(i)}$	0.1	1	10	0.1	1	10	0.1	1	10	2

Table 4.5: The values of the tested hyper-parameters $\alpha_s^{(i)}$ and $\beta_s^{(i)}$ in the prior of $\boldsymbol{\sigma}^2$ in $SVD - ADGP$ model.

$\alpha_{\theta,s,k}^{(i)}$	1	1	1	2	3	5
$\beta_{\theta,s,k}^{(i)}$	0.5	1	3	5	1	2

Table 4.6: The values of the tested hyper-parameters $\alpha_{\theta,s,k}^{(i)}$ and $\beta_{\theta,s,k}^{(i)}$ in the prior of $\boldsymbol{\theta}$ in $SVD - ADGP$ model.

We calculate the root mean square prediction error from data generating from two simulators over 50 repliacitons. The first simulator [20] is

$$y(\boldsymbol{w}, t) = m(x_1, x_2)+m(x_1, t)+m(x_4, z_4)+m(z_1, t)+m(x_5-x_2, z_2)+m(x_3, z_3), \quad (4.25)$$

where

$$\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z}), \boldsymbol{x} \in (0, 2) \times (0, 4) \times (0, 1) \times (0, 1) \times (0, 1),$$

$$z_i \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5\}, i = \{1, ..., 4\}$$

$m(\cdot,\cdot)$ is the Mccormick function defined as

$$m(x_1, x_2) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1.$$

and time points are 200 equally spaced points in the range $t \in [1.5, 3.5]$. In each replication, for the training data of size 216, the designs for quantitative factors are maximin Latin hypercube designs while the design for qualitative factors is 216-run six-level factional factorial design [15]. For the test data of size 1080, the designs for quantitative factors are random Latin hypercube designs while the qualitative factors are randomly generated by the MATLAB function $'randi'$.

The second simulator [20] is

$$y(\boldsymbol{w}, t) = 1.4h(x_1, z_1) + 2h(z_2, t) + h(x_2, z_3)t, \tag{4.26}$$

where $h(\cdot,\cdot)$ is the Holder Table function defined as

$$h(x_1, x_2) = -|\sin(x_1)\cos(x_2)\exp(|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}|)|,$$

$\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z})$, $\boldsymbol{x} \in (-1, 1) \times (2, 6)$, $z_i \in \{1, 2, 3, 4, 5\}, i = 1, ..., 3$ and time points are 200 equally spaced points in the range $t \in [2.5, 3.5]$. In each replication, for the training data of size 125, the designs for quantitative factors are maximin Latin hypercube designs while the design for qualitative factors is 125-run five-level full factorial design. For the test data of size 1250, the designs for quantitative factors are random Latin hypercube designs while the qualitative factors are randomly generated by the MATLAB function $'randi'$.

As shown in Figure 4.9 and Figure 4.11, we can see that the choice of hyper-parameters does not affect the performance of the model prediction for simulator (4.25). The results in Figure 4.10 and Figure 4.12 indicate that $Ga(1, 0.5)$ and $IG(0.1, 1)$ are the best prior for $\boldsymbol{\theta}^{(i)}$ and $(\boldsymbol{\sigma}^2)^{(i)}$, $1 \leq i \leq q$, respectively.

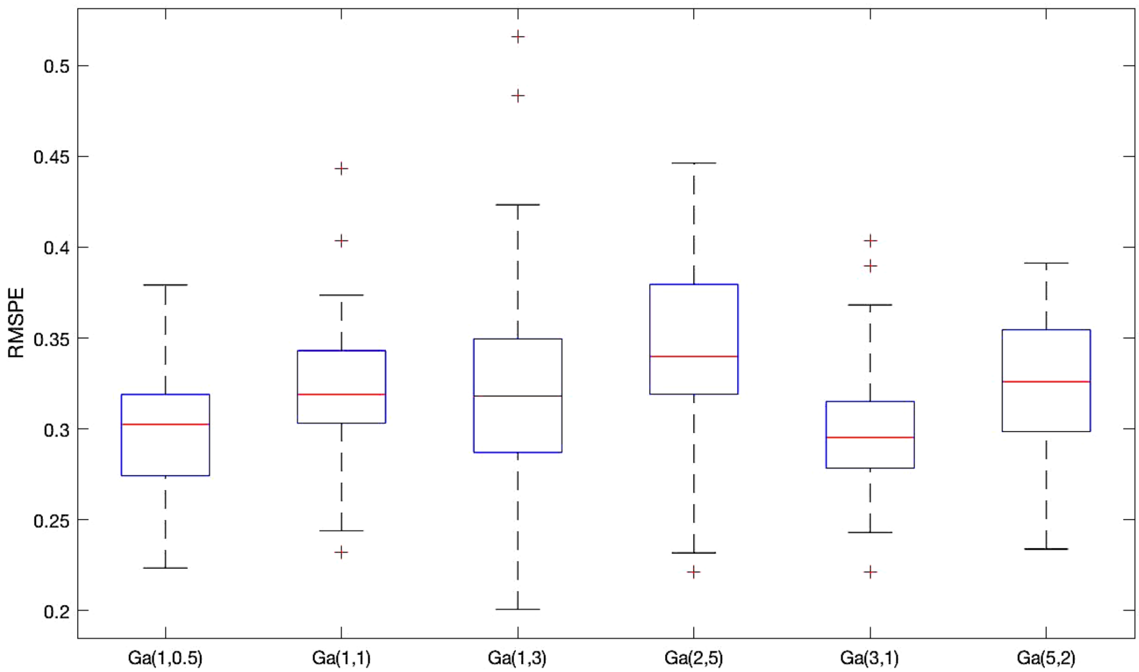

Figure 4.9: The boxplot of the RMSPEs computed from the simulator (4.25) with different hyper-parameters for $\boldsymbol{\theta}$ over 50 replications.

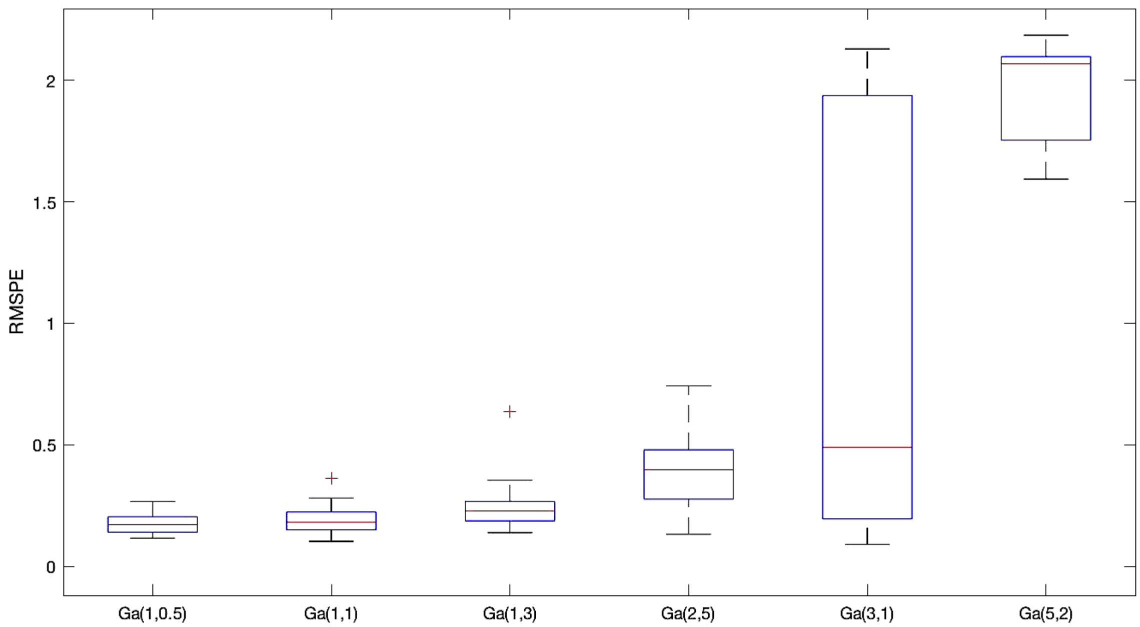

Figure 4.10: The boxplot of the RMSPEs computed from the simulator (4.26) with different hyper-parameters for $\boldsymbol{\theta}$ over 50 replications.

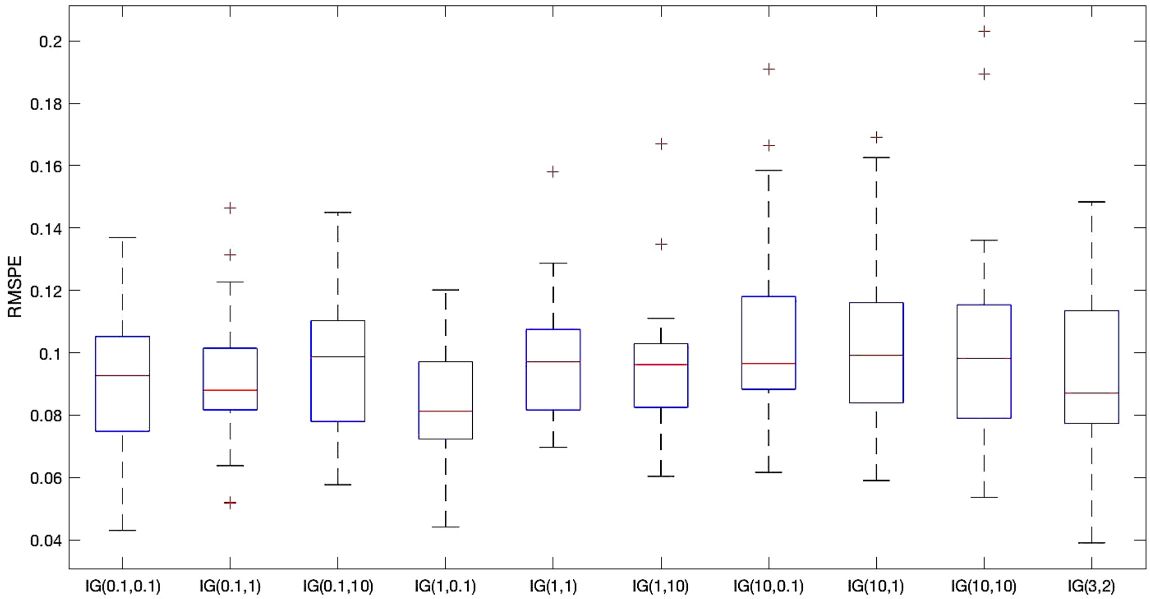

Figure 4.11: The boxplot of the RMSPEs computed from the simulator (4.25) with different hyper-parameters for $\boldsymbol{\sigma}^2$ over 50 replications.

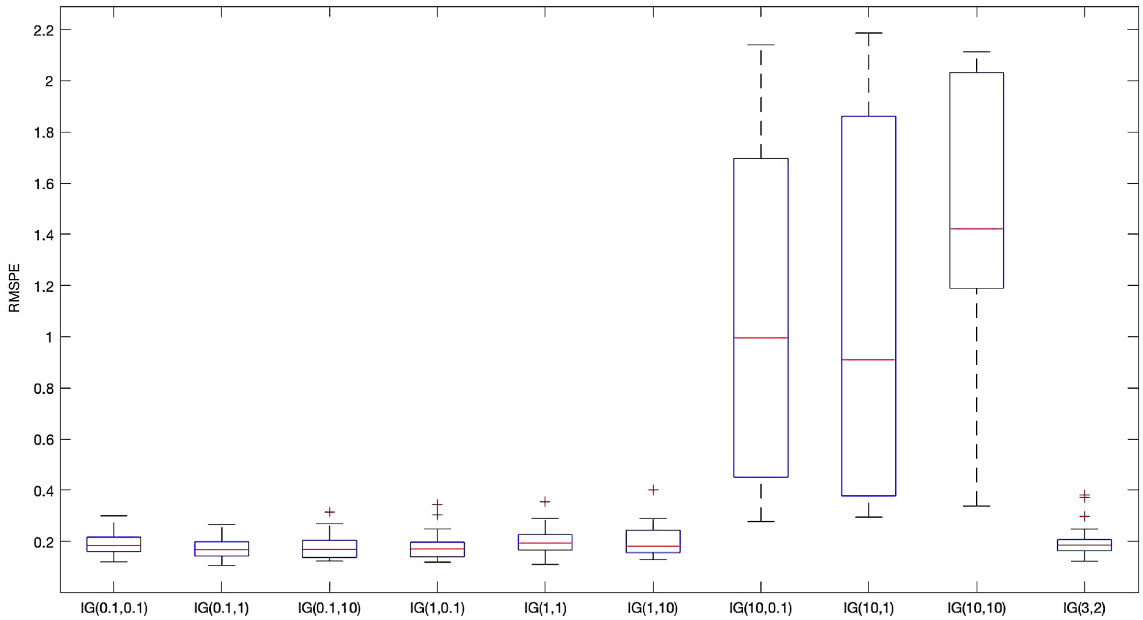

Figure 4.12: The boxplot of the RMSPEs computed from the simulator (4.26) with different hyper-parameters for $\boldsymbol{\sigma}^2$ over 50 replications.

4.3 Simulation Study

In this section, we consider three examples with different simulators to test the effectiveness of the DLM-ADGP model in Section 4.1 and SVD-ADGP model in Section 4.2. Combining the MS model in Section 2.2.3 and the additive GP model in Section 2.3, we have the ADGP model which applies the additive GP Model to each time point. The performance of the three models are evaluated by the root mean square prediction error as (3.1) over 50 replications in each example.

For the DLM-ADGP model, we use the priors as $Beta(2, 1.9)$ for the range parameter $\rho_i^{(j)}$, $IG(10, 0.1)$ for the variance component $\boldsymbol{\sigma}^2$, $Ga(0.05, 5)$ for prior marginal distribution of $\boldsymbol{v}$, $N((1,1)', diag(10, 10))$ for $\boldsymbol{\psi}$. For the SVD-ADGP model, we use the priors as $Ga(1, 0.5)$ for $\boldsymbol{\theta}^{(i)}$ and $IG(0.1, 1)$ for $(\boldsymbol{\sigma}^2)^{(i)}$, $1 \leq i \leq q$.

4.3.1 Example 1

Consider the first example with 5-dimensional inputs with $p = 2$ quantitative factors and $q = 3$ qualitative factors and time-series outputs. Data is generated from the simulator [20]

$$y(\boldsymbol{w}, t) = 1.4h(x_1, z_1) + 2h(z_2, t) + h(x_2, z_3)t, \tag{4.27}$$

where $h(\cdot, \cdot)$ is the Holder Table function defined as

$$h(x_1, x_2) = -|\sin(x_1)\cos(x_2)\exp(|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}|)|,$$

$\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z})$, $\boldsymbol{x} \in (-1, 1) \times (2, 6)$, $z_i \in \{1, 2, 3, 4, 5\}, i = 1, ..., 3$ and time points are 200 equally spaced points in the range $t \in [2.5, 3.5]$. In each replication, for the training data of size 125, the designs for quantitative factors are maximin Latin hypercube designs while the design for qualitative factors is 125-run five-level full factorial design. For the test data of size 1250, the designs for quantitative factors are random Latin hypercube designs while the qualitative factors are randomly generated by the MATLAB function $'randi'$. The results in Figure 4.13 clearly indicate that the SVD-ADGP model outperforms than the DLM-ADGP and the ADGP model.

4.3.2 Example 2

Consider the second example with 7-dimensional inputs with $p = 4$ quantitative factors and $q = 3$ qualitative factors and time-series outputs. Data is generated from

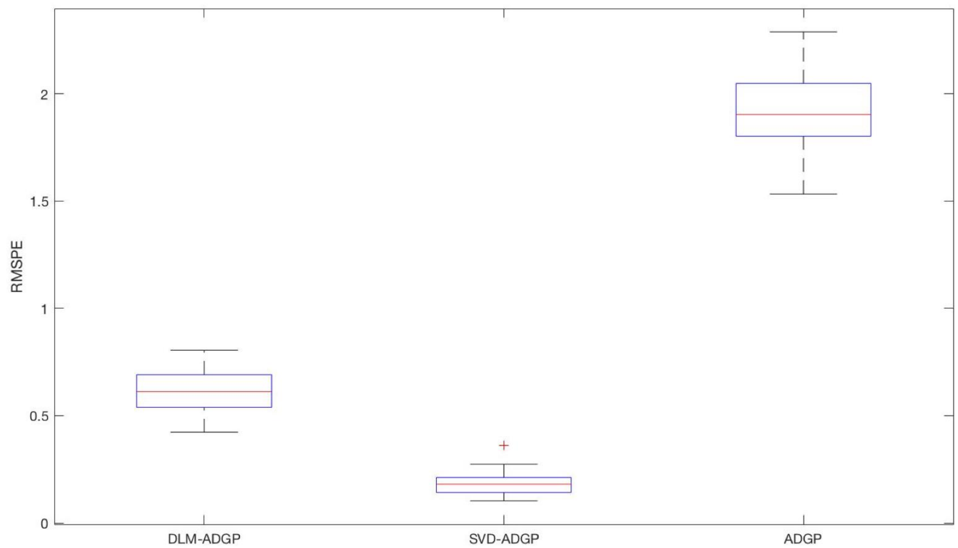

Figure 4.13: The boxplot of the RMSPEs computed from simulator (4.27) of DLM-ADGP model, SVD-ADGP model and ADGP model over 50 replications.

the simulator [20]

$$y(\boldsymbol{w}, t) = m(x_1, x_2)+m(x_1, t)+m(x_4, z_4)+m(z_1, t)+m(x_5-x_2, z_2)+m(x_3, z_3), \quad (4.28)$$

where $m(\cdot, \cdot)$ is the Mccormick function defined as (3.4), $\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z})$, $\boldsymbol{x} \in (0, 2) \times (0, 4) \times (0, 1) \times (0, 1) \times (0, 1)$, $z_i \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5\}, i = \{1, ..., 4\}$ and time points are 200 equally spaced points in the range $t \in [1.5, 3.5]$. In each replication, for the training data of size 216, the designs for quantitative factors are maximin Latin hypercube designs while the design for qualitative factors is 216-run six-level factional factorial design [15]. For the test data of size 1080, the designs for quantitative factors are random Latin hypercube designs while the qualitative factors are randomly generated by the MATLAB function $'randi'$.

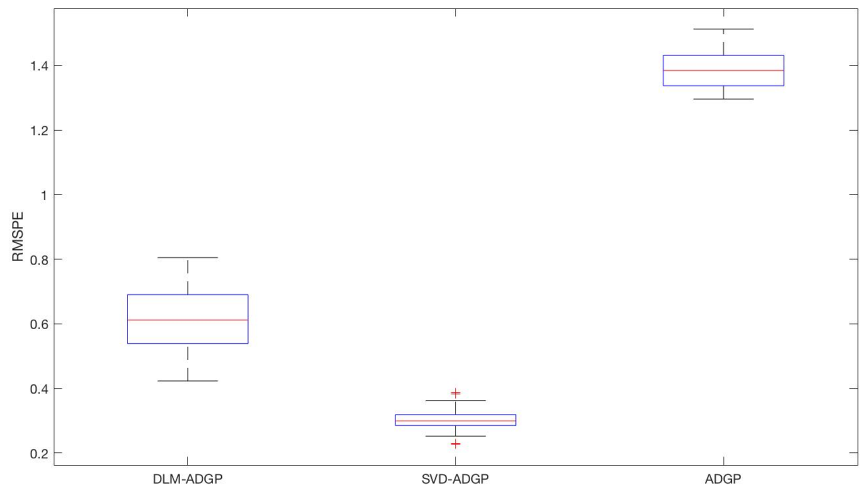

Figure 4.14: The boxplot of the RMSPEs computed from simulator (4.28) of DLM-ADGP model, SVD-ADGP model and ADGP model over 50 replications.

From Figure 4.14, it is clear that the SVD-ADGP model gives a smaller RMSPEs than the others.

4.3.3 Example 3

Consider the third example with 16-dimensional inputs with $p = 9$ quantitative factors and $q = 7$ qualitative factors and time-series outputs. Data is generated from the simulator [26]

$$\begin{aligned} y(\boldsymbol{w}, t) =& 1.5g_1(x_1 t) + 3g_2(x_2 + z_3) + 1.2g_4(x_3) + \\ & 2g_2(x_4 z_2) + 4g_2(x_7)t + 1.4g_2(z_4 + z_5) + g_2(z_6 + 1.6z_7) + \\ & 0.2g_3(x_5 + x_6) + g_1(x_8)g_2(x_9) \end{aligned} \tag{4.29}$$

where g_1, g_2, g_3, g_4 are defined in [3] as

$g_1(x) = x;$

$g_2(x) = (2x - 1)^2;$

$g_3(x) = sin(2\pi x)/[2 - sin(2\pi x)];$

$g_4(x) = 0.1sin(2\pi x) + 0.2cos(2\pi x) + 0.3[sin(2\pi x)]^2 + 0.4[cos(2\pi x)]^3 + 0.5[sin(2\pi x)]^3,$

$\boldsymbol{w} = (\boldsymbol{x}, \boldsymbol{z})$, $x_i \in (0, 1), i = 1, ..., 9, z_j \in \{0.1, 0.4, 0.9\}, j = 1, ..., 7$ and time points are 200 equally spaced points in the range $t \in (0, 1)$. In each replication, for the training data of size 81, the designs for quantitative factors are maximin Latin hypercube designs while the design for qualitative factors is 81-run three-level fractional factorial design [15]. For the test data of size 1620, the designs for quantitative factors are random Latin hypercube designs while the qualitative factors are randomly generated by the MATLAB function $'randi'$.

The results in Figure 4.15 indicates that the DLM-ADGP model is better than the others. From the three examples in this section, we can tell that the DLM-ADGP model and the SVD-ADGP model outperform the ADGP model in all examples. DLM-ADGP does well in Example 1, while SVD-ADGP shows the best performance in Examples 2-3. One possible explanation is that the time-varying auto-regressive assumption of DLM-ADGP model is suitable for simulator (4.29), but not for simulators (4.27) and (4.28).

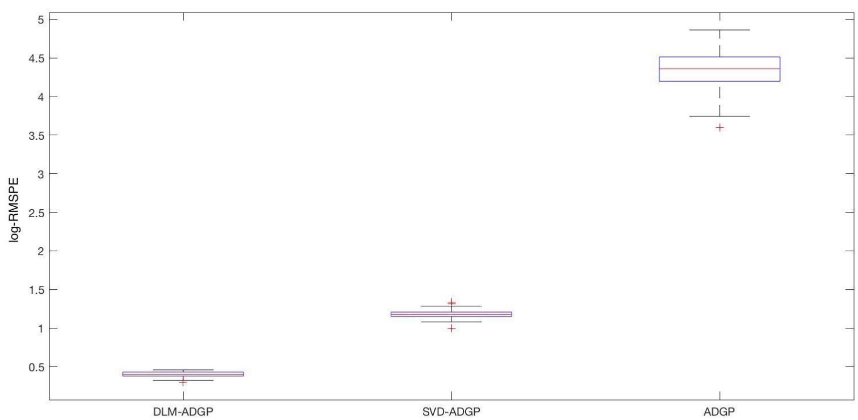

Figure 4.15: The boxplot of the RMSPEs computed from simulator (4.29) of DLM-ADGP model, SVD-ADGP model and ADGP model over 50 replications.

Chapter 5

Conclusion and Future Research

In this thesis, we make empirical comparisons of prediction accuracy of three modelling methods for dynamic computer experiments with only quantitative variables. In the examples we studied, the dynamic linear models with GP and SVD-based GP models outperform. The other contribution of this thesis is to propose two new modelling approaches for emulating dynamic computer experiments with both qualitative and quantitative inputs. The main idea of these two new modelling approaches is to incorporate qualitative factors by employing the additive correlation structure in Deng and Lin [26] in the existing two models, DLM-GP model, SVD-based GP model, for dynamic computer experiments with only quantitative variables. In addition, we have explored the different choices of the prior distributions of the unknown parameters in those two models and made the recommendation of the values of hyper-parameters. Moreover, we conduct a limited simulation study to test the effectiveness of the new models, DLM-ADGP and SVD-ADGP. We test the models with three different simulators. We have not considered the real application in this thesis and it will be our future work. The results indicate that the DLM-ADGP model and the

SVD-ADGP model outperform the ADGP model, but it seems that there is not a definite winner between the DLM-ADGP model and the SVD-ADGP model. We can see that SVD-ADGP model works better when the dimension of input is comparably small in Examples 1-2, while the DLM-ADGP model works better when the dimension of inputs is larger in Example 3. The results are similar to the results in Chapter 3, which we did the simulation study for the DLM model and SVD model with only quantitative inputs.

There are a few limitations of the research that need to be mentioned. Firstly, we only consider the additive GP model to incorporate the qualitative factors in this thesis. In fact, there are other methods in recently studies that might be considered, like the unrestrictive correlation structure proposed by [21]. Other methods to incorporate the qualitative inputs into a dynamic computer model are worth considering. Secondly, recall that the parameters needed to be estimated is $(T + q + pq + Th + \sum_{j=1}^{q} m_j(m_j - 1)/2)$ for DLM-ADGP model and $(q + pq + q\sum_{j=1}^{q} m_j(m_j - 1)/2)$ for SVD-ADGP model. Thus, the large input dimension and large levels for qualitative factors will cause computational problem for both of the two models, and the length of time series will affect the computational speed of the DLM-ADGP model. There is a need to consider the computational efficiency. Thirdly, one limitation of the two proposed modelling methods is that the input spaces of the models are all hypercubes. However, in many real-world applications, the input space is under some constrictions and it is not hypercube. How to fit the model with non-hypercube input space can be an interesting topic for research.

Bibliography

[1] Robert B.Gramacy and Daniel W.Apley. Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24:561–578, 2015.

[2] N. Bliznyuk, D. Ruppert, C. Shoemaker, R. Regis, S. Wild, and P. Mugunthan. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2):270–294, 2008.

[3] Howard D Bondell Brian J Reich, Curtis B Storlie. Variable selection in bayesian smoothing spline anova models: Application to deterministic computer codes. *Technometrics*, 51:110–120, 2009.

[4] Stefano Conti and Anthony O'Hagan. Bayesian emulation of complex multioutput and dynamic computer models. *Journal of Statistical Planning and Inference*, 140:640–651, 2010.

[5] Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Joumal of the American Statistical Association*, 86.

[6] Garrett M. Dancik. *mlegp: Maximum Likelihood Estimates of Gaussian Processes*, 2013. R package version 3.1.4.

[7] X Deng, Y Hung, and CD Lin. Design for computer experiments with qualitative and quantitative factors. *Statistica Sinica*, pages 1567–1581, 2013.

[8] Andrew Gelman et al. *Bayesian Data Analysis*. Chapman and Hall/CRC, New York, 2013.

[9] Thomas E. Fricker, Jeremy E. Oakley, and Nathan M. Urban. Multivariate gaussian process emulators with nonseparable covariance structures. *Technometrics*, 55:47–56, 2013.

[10] Andrew Gelman. Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 3:515–533, 2006.

[11] Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103.482:570–583, 2008.

[12] Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimisation problems. *eprint arXiv:1308.4008*, 2013.

[13] Krishnanand N. Kaipa and Debasish Ghose. *Glowworm Swarm Optimization: Theory, Algorithms, and Applications (Studies in Computational Intelligence)*. Springer, New York, 2017.

[14] Robert E. Kass and Larry Wasserman. The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91:1343–1370, 1996.

[15] Y.-W. Leung and Yuping Wan. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary*, 5(1):41–53, 2001.

[16] Yiu-Wing Leung and Yuping Wang. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 5(1):41–53, Feb 2001.

[17] Fei Liu and Mike West. A dynamic modelling strategy for bayesian computer model emulation. *Bayesian Anal. Bayesian Analysis*, 4.2:393–411, 2009.

[18] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21.2:239–245, 1979.

[19] Nancy J. McMillan, Jerome Sacks, William J. Welch, and Feng Gao. Analysis of protein activity data by gaussian stochastic process models. *Journal of Biopharmaceutical Statistics*, 9(1):145–160, 1999.

[20] Marcin Molga and Czesław Smutnicki. Test functions for optimization needs. *Test functions for optimization needs*, page 101, 2005.

[21] Peter Z. G. QIAN, Huaiqing WU, and C. F. Jeff WU. Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics*, 50.3:383–396, 2008.

[22] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[23] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–435, 1989.

[24] Thomas J. Santner, Brian J. Williams, and William Notz. *The Design and Analysis of Computer Experiments*. Springer, New York, 2003.

[25] X.Deng, C.Devon.Lin, K-W. Liu, and R.K.Rowe. A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics*, 53:266–273, 2011.

[26] X.Deng, C.Devon.Lin, K-W. Liu, and R.K.Rowe. Additive gaussian process for computer models with qualitative and quantitative factors. *Technometrics*, 59:283–292, 2016.