The Pennsylvania State University

The Graduate School

Department of Statistics

NEW MODELING PROCEDURES FOR FUNCTIONAL DATA IN COMPUTER EXPERIMENTS

A Thesis in

Statistics

by

Zhe Zhang

© 2007 Zhe Zhang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2007

The thesis of Zhe Zhang was reviewed and approved* by the following:

Runze Li
Associate Professor of Statistics
Thesis Advisor, Chair of Committee

James L. Rosenberger
Professor of Statistics

John Fricks
Assistant Professor of Statistics

Timothy W. Simpson
Professor of Mechanical and Industrial Engineering

Bruce G. Lindsay
Professor of Statistics
Department Head

*Signatures are on file in the Graduate School.

Abstract

New Modeling Procedures for Functional Data in Computer Experiments

Along with the rapid development of computers, computer experiment becomes more and more important in all the scientific research area. However the computer experiment usually takes long time to run for each case and requires heavy computation. Hence for large, complex systems it is not feasible to get all the values over the entire experimental space.

This thesis attempts to model the computer experiment with multivariate response or functional output using multivariate kriging and (partial) functional linear models. The whole purpose of the thesis is to propose some good interpolation models in computer experiments with multivariate or functional responses.

In chapter 3, we proposed a multivariate kriging model and multivariate functional ANOVA decomposition for computer experiments with multivariate responses. The simulation results show that multivariate version can improve the prediction up to 30%.

In chapter 4, we proposed a spatial-temporal model for the residual of functional linear model and the corresponding functional ANOVA decomposition. This model enable us to get a smooth surface which passes through each observation point. The functional ANOVA decomposition part can answers questions such as how a predictor affects the overall functional responses and how important it is. We also performed a simulation to compare the multivariate kriging and kriging with single response under this model frame. The simulation result shows that the multivariate version improves the prediction power for about 10%.

In chapter 5, we proposed a spatial-temporal model for the residual of partial functional linear model. This chapter aims to reduce the complexity of the main effect model and make it more parsimonious. We performed a simulation to compare the multivariate kriging and kriging with single response under this model frame. We observe the similar prediction power increase as in chapter 4.

Within each chapters, we applied the proposed models and estimation procedures to some real case examples. Although these examples deal with engine simulation, the proposed method can be applied to all computer experiments with multivariate or functional responses.

Key Word: computer experiment, multivariate kriging, functional ANOVA decomposition, functional linear model, semi-varying coefficient model, back-fitting algorithm

Contents

List of Tables

List of Figures

Acknowledgements

I would first like to sincerely thank Dr. Runze Li, my advisor, and Dr. Tom Hettmansperger, Dr. James Rosenberger and Dr. Steve Thompson for their help during my transition time which was the most difficult time in my life.

I would also like to thank again my advisor, Dr. Runze Li, for his guidance and patience as we worked through my research. He was always there ready to help me with any difficulties. He was always extremely supportive to me.

I would also like to thank Dr. Steve Arnold, Dr. David Hunter, Dr. Tom Hettmansperger and Dr. Bing Li for their wonderful course teaching.

I would also like to thank Dr. Agus Sudjianto in the Bank of America for his support during my summer internship. He always had many ideas whenever we reached the dead end in our research there. And he was always extremely supportive to me. He provided all the data in this thesis from his research.

And last but not least, I'd like to thank my family. They have been a great inspiration for me during all my years in school. My wife Hong and my daughter Emily sacrificed a lot for me to support my study and research. My parents have given me all the love and support with the discipline that I need for my life and I really hope I can make them proud.

This research was partially supported by National Science Foundation Grants DMS 0348869 and CCF 0430349.

Chapter 1

Introduction

Functional data are data collected over an interval of some index. For example, suppose that data for each experimental unit were collected every minute over a period of an hour, and we want to know how the factors affect the shape of the resulting curves. Functional data can be in the form of 1-dimensional data such as a curve or higher dimensional data such as a 2 or 3-dimensional image. With the advent of modern data collection devices and vast data storage space, functional data are being collected more and more in many research areas, including medical studies, substance use studies, automotive engineering, nano-scale material research and national security studies. Ramsay and Silverman (1997) present many interesting examples of functional data and introduce various statistical models to fit functional data. There are a lot excellent work have been done in the past for computer experiments. Although almost all the output of computer experiments are either multiple response or functional response, researchers treated them as a single response and model each component output one by one. This dissertation research tries to model computer experiments by treating the output as either multiple response or functional response. The work was motivated by two case studies of functional data collected at Ford motor company although the proposed models are widely suitable for all the computer experiments with multiple or functional response. For computer experiments with output has sparse sampling rate, our objective is to build a multivariate interpolation model which can consider correlation among both design variables and different components. For the computer experiments with intensive sampling rate, the output is functional response. Our objective is to build an interpolation model which can consider the correlation among design variables and the smoothness along functional direction.

With modern technology and efficient numerical methods, engineers and scientists frequently use computer experiments to study actual or theoretical physical systems in various research fields. To simulate a physical system, one needs to construct mathematical models to represent physical process. The models are often very complicated and have different levels of component such as a detailed physical model as well as a more abstract and higher level model with less detailed representation. A physics-based model may be represented by a set of equations including linear, nonlinear, ordinary and partial differential equations. Because of the complexity of real physical systems, there is usually no simple analytic formula to describe the phenomena in every detail, hence interrogattion of the models are required. In this situation, it is often difficult and even impossible to study the behavior of a computer model using traditional methods of experimentation. One approach to study complex input-output relationships exhibited by the simulation model is to construct an approximation model (also called metamodel in the literature) based on a set of limited observation data acquired by running the simulation model at carefully selected design points. Fang, et al. (2005) systematically introduced space-filling designs for computer experiments.

As modeling of computer experiments has become a popular topic in recent years, hundreds of papers have been published both in the statistics and engineering communities. Several comprehensive review papers emphasized the importance and challenge of this relatively new area (see, for example, Koehler and Owen, 1996 and

Simpson, et al. 2001). Most existing modeling procedures for computer experiments in the literature can only deal with single response computer experiments. Many interesting example of single response computer experiments can be found in the literature. We next present an example of single response computer experiments, in contrast to functional response computer experiments in Examples 1.2 and 1.3.

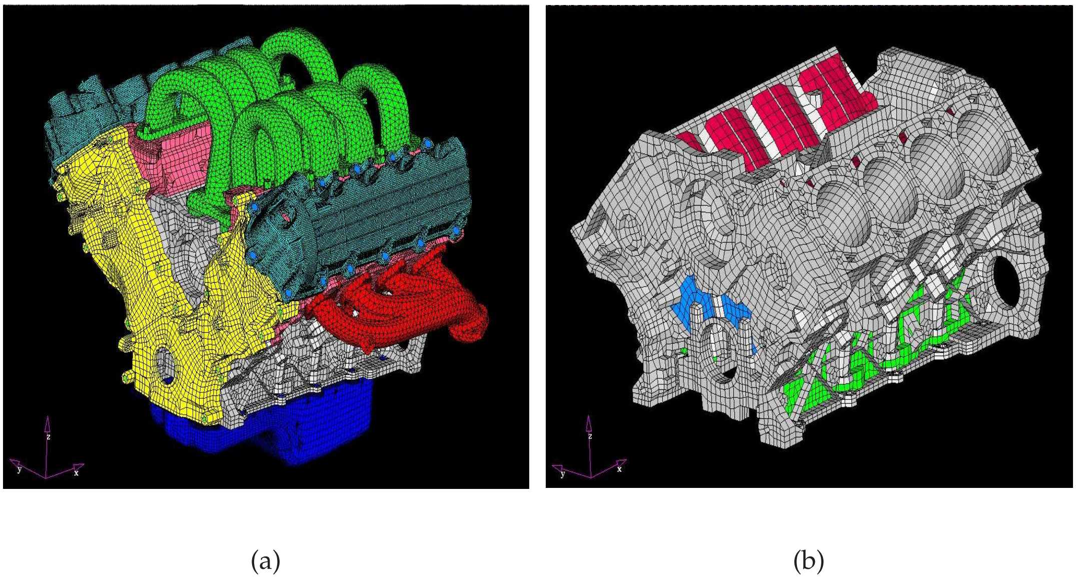
Example 1.1. *(***Robot arm***, An and Owen (2001))* This example is adapted from An and Owen (2001). Consider a robot arm with m segments. Suppose that the shoulder of the arm is fixed at the origin in the (x, y)-plane, and the length of the j-th segment of this arm is l_j, for $j = 1, \cdots, m$. The angle between the first segment and y-axis is denoted to be θ_1. For $k = 2, \cdots, m$, the angle between the $(k-1)$-th segment and the k-th segment is θ_k. By some straightforward calculation, the coordinate of the end of the robot arm in the (x, y)-plane is

$$x = \sum_{j=1}^{m} l_j \cos(\sum_{k=1}^{j} \theta_k), \text{and} y = \sum_{j=1}^{m} l_j \sin(\sum_{k=1}^{j} \theta_k),$$

and the response z is the distance $z = \sqrt{u^2 + v^2}$ from the end of the arm to the origin expressed as a function of $2m$ variables $\theta_j \in [0, 2\pi]$ and $l_j \in [0, 1]$. It is of interest to construct an approximation model $z = g(\theta_1, \cdots, \theta_m, l_1, \cdots, l_m)$ for the robot arm with m segments. The movement trajectory of a robot arm has been used as a classic illustrative example in the literature of neural network.

1.1 Motivating Example

With the development of modern computer technology, outputs of computer models are often presented as a higher dimensional image. Computer experiments with functional responses have increased in complexity and resulted in today's sophisticated three-dimensional computer models. However, there is little literature on modeling of computer experiments with functional response, and modeling computer experiments is typically limited to point outputs. The work was motivated by the following two examples, whose detailed description and brief analysis are given in Fang, et al. (2005). In this dissertation research, our target is to provide a more sophisticated approach to analyze functional responses.



(a) (b)

Figure 1.1. Engine models for Example 1.2: (a) engine system finite element model, and (b) engine block finite element model.

Example 1.2. (*Engine Noise, Vibration and Harshness*) Computer models are frequently utilized in the design of engine structure to reduce radiated noise. The model includes multi-body dynamic simulation to estimate acting mechanical forces during engine operations from the cranktrain and valvetrain, as well as combustion forces. Together with the finite element structural model of the engine component or system (see Figure 1.1), this model is applied to evaluate structural response during operating conditions as a function of engine RPM and frequency. The vibro-acoustic relationship between the engine vibrations and the acoustic pressure field may then be applied to calculate the radiated engine noise (see, for example, Gérard, et al. 2001).

To optimize the design of the cylinder block, 17 design variables (e.g., bulk-

Table 1.1. Name of Design Variables

Var.	Name	Var.	Name
x_1	Bulkhead thickness	x_{10}	Side wall ribbing
x_2	Oil pan rail thickness	x_{11}	RFOB ribbing
x_3	Skirt wall thickness	x_{12}	FFOB ribbing
x_4	Main cap thickness	x_{13}	Valley wall ribbing
x_5	Liner thickness	x_{14}	Bearing beam
x_6	Valley wall thickness	x_{15}	Dam with value 1
x_7	Bore wall thickness	x_{16}	Dam with value 2
x_8	Valley plate thickness	x_{17}	Den
x_9	Skirt ribbing	x_{18}	Young modulus

head thickness, oil pan rail thickness, valley wall ribbing, etc.; a detailed list is given in Table 1.1) were chosen. A uniform design with 30 runs (Fang, et al. (2005)) was used for the experiment and is presented in Table 1.2. In this example, the selected response variable is the structural response of the oil pan flange in terms of acceleration at various engine RPMs (e.g., 1000, 2000,..., 6000 rotation per minute(RPM)), which directly influences the radiated noise through the vibro-acoustic relationship. The outputs are presented in Table 1.4 and are depicted in Figure 1.2.

This is a typical example of functional response with a sparse sampling rate. The output was collected over different RPMs, and there are only 6 outputs in total collected for each design condition. This response is similar to that in a growth curve model in statistical literature (See, for example, Pan and Fang, 2004). Although one may view such output as multi-response, we refer to it as a functional response because the data could be collected over the whole interval of RPM, and the response is several outputs of one variable rather than outputs of several variables.

Table 1.2. Design of Engine Noise Vibration and Harshness

Run #	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	25.5	9	7.5	22.5	5.5	9	5.5	8.5	1
2	25.5	9	6.5	25.5	3.5	13	7	13	0
3	18.5	21	7.5	18.5	3.5	11	7	13	1
4	22.5	13	8.5	25.5	2	13	5.5	8.5	0
5	22.5	17	5	20.5	2	6	9	3.5	0
6	24.5	9	6.5	22.5	3.5	4	7	13	0
7	22.5	13	5	25.5	5.5	13	5.5	6	1
8	24.5	17	8.5	20.5	3.5	13	9	6	0
9	18.5	17	5	20.5	5.5	9	5.5	3.5	1
10	20.5	9	9	18.5	3.5	11	5.5	3.5	1
11	18.5	9	9	24.5	3.5	11	9	3.5	1
12	20.5	13	6.5	24.5	2	6	9	8.5	1
13	25.5	13	7.5	22.5	5.5	13	9	8.5	1
14	25.5	13	9	25.5	2	4	7	13	0
15	24.5	9	8.5	18.5	2	6	9	10	1
16	18.5	21	5	20.5	5.5	9	5.5	8.5	0
17	22.5	21	6.5	25.5	2	11	5.5	10	1
18	24.5	23	9	22.5	3.5	11	9	10	0
19	25.5	21	7.5	24.5	2	4	7	3.5	0
20	20.5	17	9	22.5	5.5	9	9	13	1
21	18.5	23	7.5	24.5	5.5	11	9	13	0
22	24.5	23	8.5	20.5	2	13	7	3.5	0
23	20.5	23	6.5	25.5	2	6	5.5	10	0
24	22.5	23	8.5	24.5	3.5	6	7	6	1
25	24.5	21	5	20.5	2	9	7	6	1
26	22.5	17	6.5	18.5	3.5	4	9	8.5	0
27	20.5	13	9	22.5	5.5	4	7	10	1
28	20.5	21	8.5	24.5	5.5	6	5.5	6	0
29	25.5	17	5	18.5	3.5	4	5.5	6	0
30	18.5	23	7.5	18.5	5.5	9	7	10	1

Table 1.3. Design of Engine Noise Vibration and Harshness (continued)

Run #	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	$\{x_{15}, x_{16}\}$	x_{17}	x_{18}
1	0	1	0	1	0	3	2.91	77.49
2	1	1	1	1	1	2	2.77	70.11
3	1	0	1	0	0	1	2.91	77.49
4	0	0	0	0	1	3	2.77	77.49
5	1	1	1	1	1	1	2.77	77.49
6	0	0	1	0	1	1	2.63	73.8
7	1	1	0	0	1	2	2.77	77.49
8	1	0	0	1	0	3	2.63	70.11
9	1	0	1	1	0	3	2.63	73.8
10	0	1	1	0	1	2	2.77	77.49
11	0	0	0	0	1	3	2.91	73.8
12	1	0	0	0	0	1	2.91	73.8
13	1	1	1	0	0	1	2.77	77.49
14	1	1	0	1	0	2	2.77	77.49
15	0	0	1	1	0	2	2.63	73.8
16	0	0	0	1	1	2	2.91	70.11
17	0	1	1	1	0	2	2.63	77.49
18	0	0	0	1	0	1	2.91	73.8
19	1	0	1	0	1	3	2.77	70.11
20	0	0	1	0	0	2	2.63	70.11
21	1	0	0	0	0	2	2.63	73.8
22	0	1	0	1	1	2	2.91	73.8
23	1	1	1	0	1	3	2.77	73.8
24	1	1	0	1	1	3	2.63	70.11
25	0	1	0	0	0	1	2.91	70.11
26	1	0	1	1	1	1	2.91	70.11
27	1	0	1	1	1	3	2.77	77.49
28	0	1	0	0	0	1	2.63	70.11
29	0	1	1	0	0	3	2.91	73.8
30	0	1	0	1	1	1	2.63	70.11

Table 1.4. Response of Engine Noise Vibration and Harshness

Run #	1000	2000	3000	4000	5000	6000
1	0.09	0.39	1.01	2.84	8.51	14.73
2	0.11	0.46	1.23	3.46	9.39	18.32
3	0.09	0.38	1.03	3.15	8.76	14.89
4	0.10	0.42	1.11	3.35	9.09	14.76
5	0.12	0.49	1.27	3.95	10.76	18.16
6	0.11	0.46	1.20	3.72	11.08	17.17
7	0.10	0.42	1.14	3.28	8.95	14.73
8	0.09	0.41	1.07	3.16	9.02	15.92
9	0.09	0.39	1.01	3.09	8.42	13.20
10	0.10	0.42	1.12	3.52	9.76	16.02
11	0.10	0.41	1.10	3.41	7.92	14.89
12	0.11	0.44	1.19	3.81	9.60	17.33
13	0.10	0.41	1.13	3.10	8.62	13.83
14	0.10	0.43	1.17	3.46	9.76	17.37
15	0.10	0.45	1.15	3.48	9.39	16.54
16	0.10	0.43	1.14	3.67	8.17	14.63
17	0.09	0.38	1.01	3.17	8.31	12.32
18	0.09	0.40	1.05	3.09	8.95	14.77
19	0.10	0.42	1.10	3.70	9.06	18.14
20	0.10	0.42	1.17	3.19	9.46	17.35
21	0.09	0.40	1.06	3.13	8.52	14.09
22	0.10	0.44	1.16	3.41	8.92	16.99
23	0.09	0.39	1.02	3.16	8.77	14.81
24	0.09	0.39	1.04	3.24	8.54	15.47
25	0.11	0.44	1.20	4.04	10.30	19.03
26	0.11	0.48	1.27	4.33	9.72	18.48
27	0.09	0.39	1.07	3.15	8.74	14.71
28	0.10	0.46	1.23	3.46	9.62	19.06
29	0.10	0.40	1.13	3.63	9.79	18.72
30	0.10	0.41	1.14	3.50	10.01	17.42

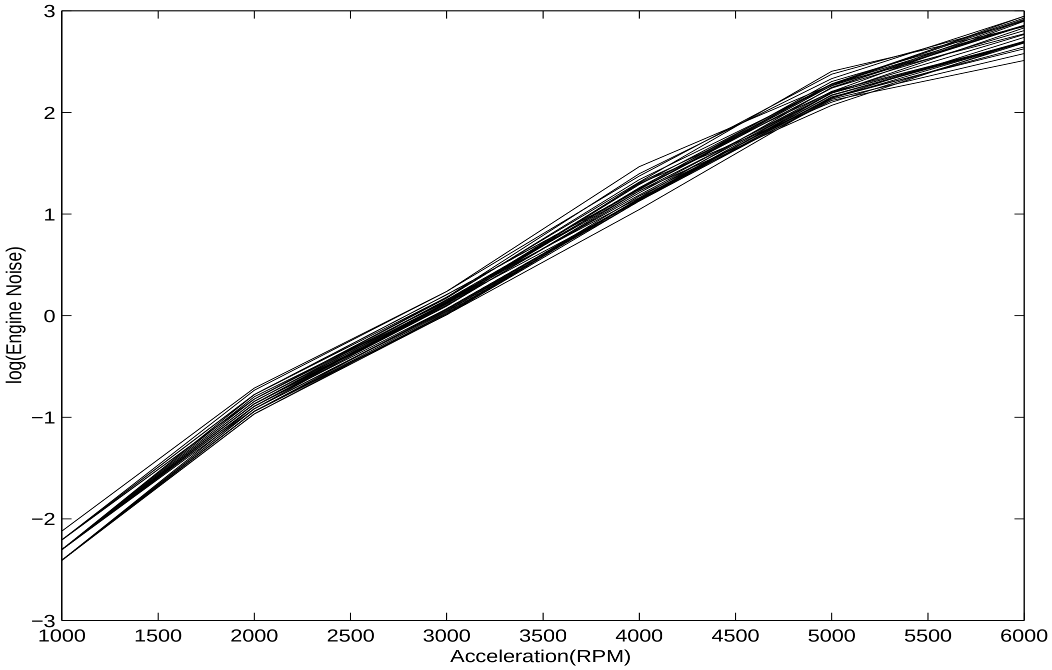


Figure 1.2. Plot of logarithm of response for Example 1.2:

Example 1.3. (*Valvetrain*) The valvetrain system is one of the most important elements of internal combustion engines for the delivery of desired engine performance in terms of horsepower/torque, noise and vibration, fuel economy, and emissions. Design optimization is performed to maximize performance capability while maintaining the durability of the system. During the optimization process, engineers synthesize the design of components in the valvetrain system (e.g., cam profile, spring, etc.) to meet the intended performance target. To achieve such goals, various computer-aided engineering models are simultaneously employed to achieve the best balance of performance and durability attributes. Multi-body dynamic computer models are commonly employed to optimize the dynamic behavior of a valvetrain, especially at high engine speeds, where the spring behavior becomes highly nonlinear (see Figure 1.3 below) in an internal combustion engine (Philips, Schamel and Meyer (1989). The model can be used to study typical valvetrain system characteristics in terms of durability, noise, and valve sealing. Detailed three-dimensional geometric information as well as other physical properties such as valve seat, spring (spring surge and coil clash), camshaft torsional and bending, cam-follower hydrodynamic, and lash adjuster dynamics are included in the models. The model is very crucial for guiding valvetrain performance optimization while maintaining a stable valvetrain dynamic behavior. One indicator of a stable valvetrain is that the valve movement must follow a prescribed motion determined by the camshaft profile. At high speed valvetrain operations, however, this may not be the case, as the effect of inertia on the dynamic behavior becomes prominent. This phenomenum is especially crucial during valve closing to minimize valve bounce. Figure 1.4 shows the motion errors of the valve compared to the prescribed motion by the camshaft for the first four designs listed in Table 1.5. A perfectly stable valve train should have a straight line or zero error throughout the crank angles. From Figure 1.4, we can see that the variability of the motion errors varies for different design configurations.

In this situation, engineers attempt to minimize motion errors by finding the best level-combination of the factors, such as cylinder head stiffness, rocker arm stiffness, hydraulic lash adjuster, spring, cam profile, etc. To achieve this goal, one is interested in understanding the effects of each design variable on the motion errors. In this case, the response variable (i.e., amount of motion error) is defined over a crank angle range, instead of a single value; thus, it is considered a functional response.

Table 1.5 depicts an experimental matrix which was applied to the computer model to minimize the motion error, particularly to minimize valve bounce during valve closing. This is a typical example of functional response with intensively sampling rate. To build up a parsimonious metamodel, nonparametric smoothing techniques should be used to construct a good predictor for the functional response.

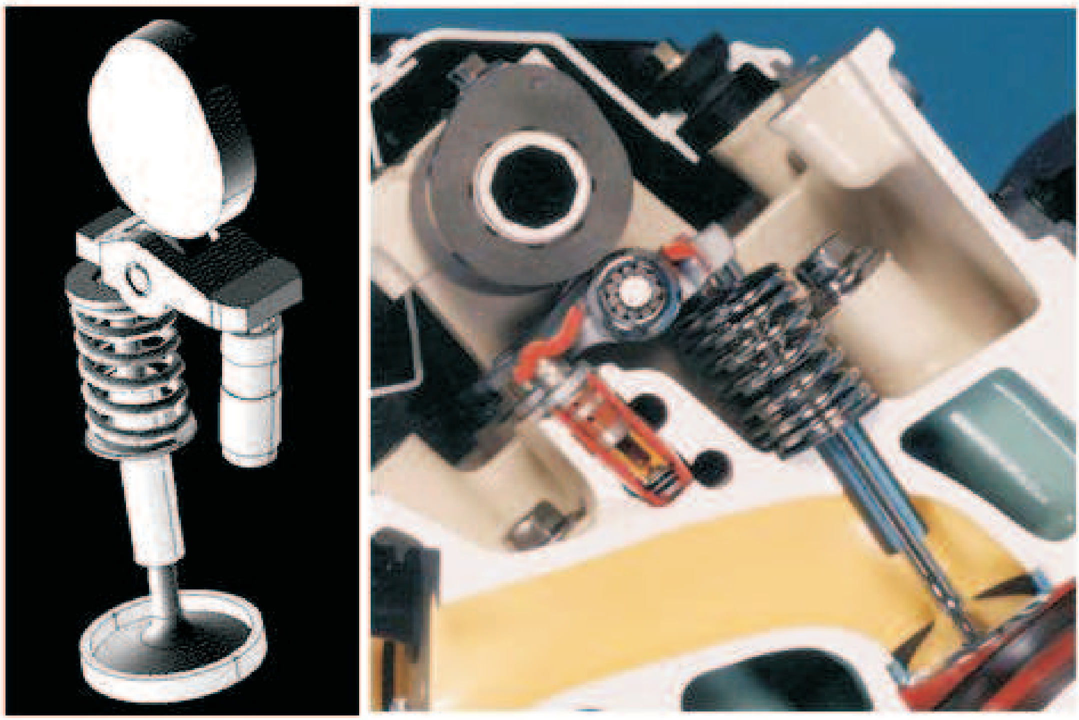


Figure 1.3. Roller finger follower valvetrain system.

1.2 Contribution of this Dissertation

As mentioned before, there is few existing works on modeling functional response computer experiments. Li, Sudjianto and Zhang (2005) have done pioneering work on modeling computer experiment with a functional response by using functional linear models. A general framework of modeling functional response computer experiments is given in Chapter 7 of Fang, Li and Sudjianto (2005). In this dissertation, we will develop new modeling procedures for functional response computer experiments.

In Chapter 3, we will propose multivariate Gaussian kriging models for sparse functional response computer experiments, of which Example 1.2 is a typical example. The newly proposed multivariate Gaussian kriging models may be directly

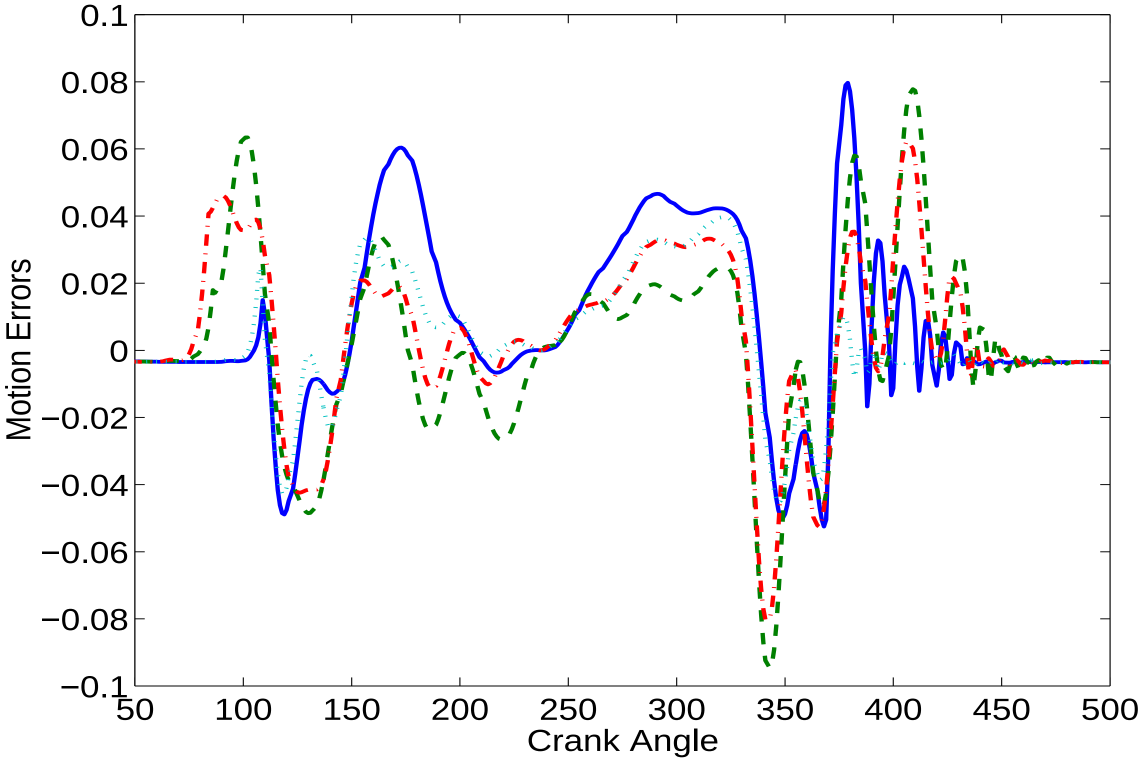


Figure 1.4. Valve motion errors of the first four valvetrain designs listed in Table 1.5.

applied for multiple response computer experiments. Using the theory of the matrix normal distribution, we propose a new estimation procedure for the multivariate Gaussian kriging models and derive the best linear unbiased prediction for the sparse functional response computer experiments. We further proposed an algorithm to maximize the likelihood function of multivariate Gaussian kriging models and discuss issues related to implementation the proposed algorithm. To interpret the resulting models, we further extend the functional ANalysis Of VAariance (ANOVA) decomposition to multivariate Guassian kriging models. To conduct functional ANOVA decomposition, we derive the closed forms for the integrals involved in the functional ANOVA decomposition. All proposed modeling procedures are validated by a simulation study. We further compare the multivariate kriging model with the existing univariate kriging model for single response computer experiments

Table 1.5. Design for Valvetrain Experiment

Head	RA	Lash	Cam	Clearance	Spring	Ramp
Stiffness	Stiffness	Adjuster	Phasing		Height	
1	1	1	1	1	1	1
2	1	1	1	1	2	3
1	2	1	1	2	1	3
2	2	1	1	2	2	1
1	1	2	1	2	2	2
2	1	2	1	2	1	2
1	2	2	1	1	2	2
2	2	2	1	1	1	2
1	1	1	2	2	2	2
2	1	1	2	2	1	2
1	2	1	2	1	2	2
2	2	1	2	1	1	2
1	1	2	2	1	1	3
2	1	2	2	1	2	1
1	2	2	2	2	1	1
2	2	2	2	2	2	3

by simulation studies. We found that the proposed multivariate kriging models outperforms the univarate kriging models. For example, the multivariate kriging models may reduce 30% prediction error when noise level is high. As an illustration, we give a detailed analysis of data presented in Example 1.2 in this chapter.

Chapters 4 and 5 are devoted to developing new modeling procedures for functional response computer experiments with high sampling rate. Example 1.3 is a typical example of this kind of computer experiments. We propose spatial-temporal model for functional response computer experiments. The proposed spatial-temporal model is a natural extension of the Gaussian kriging model for functional response computer experiments. We further propose functional linear models to represent the overall trend in the spatial-temporal model in Chapter 4, while partially functional linear models are used to estimate the overall trend in the spatial-temporal model in Chapter 5. Due to the special data structure of functional response computer experiments, we propose an estimation procedure for functional linear models in Chapter 4, and a back-fitting algorithm for partially functional linear models in Chapter 5. The error process in the spatial-temporal model is assumed to be a Gaussian process with both spatial and temporal indices. We extended the estimation procedure of kriging models for the error Gaussian process, and derive the best linear unbiased prediction. We further extended the functional ANOVA decomposition for the proposed spatial-temporal models. A simulation study was conducted to validate the proposed modeling procedure. Compared with the method proposed in Fang, Li and Sudjianto (2005), our proposed method yields more accurate prediction. The gain varies from 8% to 20%. A detailed analysis of Example 1.3 are used to illustrate the proposed methodology.

1.3 Organization of this Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 presents the literature review for existing modeling techniques for computer experiments. Specifically, Section 2.2 presents existing models for computer experiments. Kernel regression and local polynomial regression techniques are presented in Section 2.3, The functional linear models are introduced in Section 2.4, and the partially functional linear models are briefly described in Section 2.5. In Chapter 3, we propose a model for functional response with a sparse sampling rate and develop an estimation procedure for the proposed model. In Chapter 4, we propose a modeling procedure for computer experiments with functional response with an intensive sampling rate. In Chapter 5, we propose a partially functional linear model and the associated spatial-temporal model for computer experiments, which is a more parsimonious model than the functional linear model. Chapter 6 presents a summary of this dissertation work, and propose some future research work.

Chapter 2

Literature Review

Similar to physical experiments, computer experiments are performed to study a certain problem under a certain set of controlled experimental variables (input variables). Due to the rapid development of computer power at almost a fixed price, researchers conduct more and more computer experiments. In the early stages of product development, researchers prefer computer experiments over physical experiments because it is less expensive and less time-consuming. At the middle stage of product development, researchers conduct both computer experiments and physical experiments. For example, when there are some variables which are beyond control by the experiment, researchers want to use computer experiments to simulate the situation which is the same as the physical experiment and then to check if the computer simulation result is the same or at least similar with the real experimental result. This is called verification. Then researchers will vary the 'out of control variables' in the code and conduct the computer experiment to examine the effect of those variables.

In various research areas, scientists rely on computer experiments. Typically, in computer experiments, the same input yields the same output as there is no random error involved in the experiments. If the model is not complicated or if the problem is simple enough, then the computer experiments can be very fast.

The disadvantages for computer experiment include that it is not real in the sense that we probably can never find the true mechanism in the real process. When we model it in the computer experiment, the accuracy of the model is the biggest question. It is not real (or at least not as real as the real experiments) also because it cannot take into account environmental factors as in a real experiment and it can not take the variation of the value for the input variable into account.

To deal with the first problem, one needs better computer models and hence models become more and more complicated. This results in the need for more computer resources. In reality, one always pushes the model to the limit of the computer resources, which results in the computation time remaining almost unchanged with the rapid increase of CPU speed and capacity. Hence, to solve large and complicated problems, the computation is always time-consuming. Usually one run on a supercomputer for a complicated problem takes weeks or a month. The benefit of this is that the computational result is more accurate and realistic. But still the expense of the computation eliminates the possibility of running it thousands of times in order to explore in detail the importance of the input variables and to examine the model better.

Statisticians may play important roles in dealing with these problems in computer experiments because they can handle the uncertainty by using statistical methods. In a computer experiment, there are always some input variables and the experiment result will have some output. Let $\mathbf{X} = (X_1, X_2, \cdots, X_p) \in R^p$ denote the input variable vector for the computer code, where X_j is the j-th component in the input variable $\mathbf{X}$. Let $\mathbf{Y} = (Y_1, Y_2, \cdots, Y_q) \in R^q$ denote the computer output and Y_j is the j-th component of the output. Let $f(\mathbf{X})$ stand for the computer code with input variable $\mathbf{X}$, then the model for the computer experiment is

$$\mathbf{Y} = f(\mathbf{X}). \tag{2.1}$$

Often we can normalize each of the components X_j in the input variable and hence we can treat the input variable $\mathbf{X} \in [0,1]^p$. The number of input variables, p, the model function, $f(\mathbf{X})$, and the time to compute the function $f(\mathbf{X})$ are three important factors in these computer experiments. Sometimes, p is very small; sometimes, there are over fifty input variables for the code. The model function sometimes is very simple: the code only consists of several lines. However, in general, there are tens of thousands of lines of code which is a fairly complicated model. The computing time ranges from several milliseconds on a PC to weeks of CPU hours on a parallel cluster machine.

There is no random error associated with the model (2.1). So the same input value $\mathbf{X}$, the same computer experiment will give the same output $\mathbf{Y}$. All the computer experiments are deterministic for given input values. So can a statistician deal with this kind of problem using statistical method? Where does the uncertainty come in?

To answer these questions, we need to carefully consider the real meaning of computer experiments. There are several types of the input variables. The most frequently seen variables are control variables whose values can be set accurately by the scientist to control the output of the process. The second type of input variables are environmental variables which can not be set accurately in a physical experiment and hence vary much more. For example, in the Computational Fluid Dynamics (CFD) study of the flow through a pipe, the pipe's diameter is a control variable because it will not change during the experiment. On the other hand the flow velocity and the flow flux are environmental variables because although they can be controlled within a small range, they will not be constant throughout the entire real experiment. So from a statistical point of view, these environment variables will be treated as random variables with some distributions.

2.1 Goals for computer experiment

There are many situations in which an accurate approximate model (also known as metamodel) for computer experiments is required. For most situations, the time and resources which are required by the computer simulation code (2.1) is enormous and expensive. People cannot afford to run the code many times to get the output at a dense of input variable points. And we need to do prediction, optimization, robust design and sensitivity analysis. We need a metamodel which is easy and quick to compute. Another situation in which we need metamodeling is when the computer model $f(\mathbf{X})$ is known but has a very complicated form, and we need to understand this model in depth. Then, we can use metamodeling to fit the data from easy to compute model $g(\mathbf{X})$ and obtain a model with a simpler form. We usually assume $f(\mathbf{X})$ is a smooth function since it often simulates some physical process and physical processes are generally smooth.

The quality of the model can be measured using various criteria, and one of them is overall bias. Bias is defined to be the difference between the true model and expectation of the metamodel. The overall bias can be defined on an L_p space, and the most often used one is on L_2 space, which is also called **integrated square error (ISE)**

$$\int_{\chi}(f(\mathbf{x}) - g(\mathbf{x}))^2 w(\mathbf{x}) d\mathbf{x}, \tag{2.2}$$

where χ is the region of input variables, $w(\mathbf{x})$ is weighted function which defines the relative importance of each point in the region χ. In computer experiments, since there is no measurement random error, ISE is equivalent to the usual **prediction error (PE)**. By minimizing ISE (PE in computer experiments), we can obtain the least squares estimator for the metamodel (2.4) below.

Actually ISE is the square of the L_2 distance between $g(\mathbf{X})$ and $f(\mathbf{X})$. Unfortunately, we can not calculate ISE in most situations because it is too expensive to evaluate $f(\mathbf{X})$ at all untried input value. Another measure is the **mean integrated square error (MISE)**

$$\begin{aligned} MISE &= E\int_{\mathcal{X}}(f(\mathbf{x})-g(\mathbf{x}))^2 d\mathbf{x} = \int_{\mathcal{X}} E(f(\mathbf{x})-g(\mathbf{x}))^2 d\mathbf{x} \\ &= \int_{\mathcal{X}}(Eg(\mathbf{x})-f(\mathbf{x}))^2 d\mathbf{x} + \mathrm{Var}\{g(\mathbf{x})\} \qquad (2.3) \\ &= bias^2\{g(\mathbf{x})\}d\mathbf{x} + \mathrm{Var}\{g(\mathbf{x})\}, \end{aligned}$$

where we assume the expectation and integration can be exchangeable. From this formula, we can see the MISE consists of two parts, variance and bias. Generally, we will not be able to minimize variance and bias simultaneously. Instead, the variance will increase if we try to decrease the bias and vice verse. Hence, we need to find a set of estimators which can minimize the total MISE.

We often want to find the global maximum/minimum for the computer model (2.1). If the time and resources required by the computer code is fairly long and expensive, then we cannot afford to run the model (2.1) over a set of dense grid points in order to find the maximum/minimum. As an alternative, we can search over all the dense grid points using the metamodel which is easy and quick to compute. If the bias and variance of the metamodel is small enough, then we can use the maximum/minimum of metamodel to approximate the corresponding extreme value of the original computer model.

The optimization procedure involves multiple responses as the objective function to be maximized/minimized with some kind of constraints to be enforced. Metamodeling together with nonlinear iteration method can be utilized to solve this kind of constrained optimization problem.

When input variables contain some environmental variables, we will need to do sensitivity analysis to see whether the computer model is sensitive to the input values. In other words, how much variation will the response have if the input variable changes a bit? This kind of analysis is one of the main interests in the decision making process such as engineering design. The most often used measurement of the proportion of variation of response y explained by variations of input variables **x** (MacKay (1995)) is

$$CR_i = \frac{\text{Var}\{E\{y|x_i\}\}}{Var\{y\}}$$

for all the components of input variable **x** where CR_i is called the correlation ratio. This can be easily obtained by using the easy to compute metamodel. This is especially attractive when the computer code $f(\mathbf{x})$ is expensive to run. Chapter 6 of Fang etc (2005) gives a thorough discussion on sensitivity analysis.

2.2 Models for computer experiment with a single response

Once the computer experiment (computer simulation code) is fixed, the computer model (2.1) is also fixed. By 'fixed', we mean for the same input parameter **X**, the model (2.1) will give out the same output **Y**. This is a deterministic model without additive random error on the response. Typically, we approximate the computer experiment model (2.1) by

$$f(\mathbf{X}) \approx g(\mathbf{X}) + \eta(\mathbf{X}) \tag{2.4}$$

where $g(\mathbf{X})$ is an approximation function whose expectation is equal to $f(\mathbf{X})$. Depending on different approaches, $\eta(\mathbf{X})$ have different meanings. One calls this kind of model a 'model of the model' or **metamodel** (Kleijnen (1987)).

In the regression approach, we will consider $\eta(\mathbf{X})$ as the error between the mean function $g(\mathbf{X})$ and the computer model $f(\mathbf{X})$. The easiest approach is linear regression, in which one takes $g(\mathbf{X}) = Z(\mathbf{X})\beta$. Then the least square estimator for the coefficients $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}_{LS} = (\int Z(\mathbf{X})'Z(\mathbf{X})dF)^{-1} \int Z(\mathbf{X})'\mathbf{Y}dF \tag{2.5}$$

where F is the distribution of **X**. Then

$$IMSE = \int (\mathbf{Y} - Z(\mathbf{X})\boldsymbol{\beta})^2 dF$$

The integration in (2.5) can be estimated by using the usual average over the observed data information. Then (2.5) can be written as

$$\hat{\boldsymbol{\beta}}_{LS} = (\frac{1}{n}\sum_{i=1}^{n} Z(\mathbf{x}_i)'Z(\mathbf{x}_i))^{-1}\frac{1}{n}\sum_{i=1}^{n} Z(\mathbf{x}_i)'Y_i \tag{2.6}$$

If the computer model $f(\mathbf{X})$ is approximately linear in **X**, we can expect that linear regression will perform well. Otherwise, linear regression will generate a large bias. Polynomial regression, spline methods, local modeling, and neural networks are all regression methods which attempt to improve this.

In the literature of modeling computer experiments, the most common method is to interpolate the outputs over the design points. Therefore metamodels are often referred to as surrogates, emulators, or low-fidelity models (as opposed to the high-fidelity computer experiment model) (Bates et al. (2003)). It is well known in polynomial series regression that when the number of predictors equals the number of observations, then the mean function will pass through each observational point. A kriging approach is another one of these models. Since in this thesis, I use the extension of single variable kriging model and local modeling technique, the next couple sections introduce the background for these models in detail.

2.3 Kriging Model

The kriging model is an interpolation model which is widely used in many areas (Journel and Huijbregts (1978), Ripley (1981)). It was originated in geostatistics in the 1960's by Georges Matheron in Paris who applied it to problems in the mining industry(Matheron (1963)). Several empirical studies have proved its superiority over other interpolating techniques such as splines (Laslett (1994)). Kriging is now widely used in many fields including spatial statistics (Cressie (1986), Cressie (1993)) and computer experiments (Sacks et al. (1989b)). The kriging model interpolates data at the observation point, and the variance becomes larger as the prediction point moves away from the observation points. It can be interpreted from a Bayesian point of view.

In kriging, the mean function $f(\mathbf{X})$ in model (2.1) can be modeled as two parts

$$\mathbf{Y}(\mathbf{x}) = Z(\mathbf{x})\boldsymbol{\beta} + \Phi(\mathbf{x}). \tag{2.7}$$

The first part is the general linear model, $Z(\mathbf{x})$ are known fixed basis functions, $\boldsymbol{\beta}$ are unknown coefficients which we need to estimate. The second part $\Phi(\mathbf{x})$ is a stationary Gaussian random function which has multivariate normal distribution with $E[\Phi(\mathbf{X})] = 0$ and covariance

$$\text{Cov}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) = \sigma^2 R(\|\mathbf{x}_i - \mathbf{x}_j\|), \tag{2.8}$$

where $R(\|\mathbf{x}_i - \mathbf{x}_j\|)$ is the correlation coefficient between two points $\mathbf{x}_i$ and $\mathbf{x}_j$.

Denote the computer output from (2.1) as $(\mathbf{x}_i, y_i), i = 1, \cdots, n$, let $\mathbf{Y} = (y_1, \cdots, y_n)'$ be a realization of the random variable. Let the (i, j)-th element of the matrix Σ as

$$\Sigma_{ij} = \sigma^2 R(\|\mathbf{x}_i - \mathbf{x}_j\|),$$

then Σ is the variance-covariance matrix for $\mathbf{y}$. Let $\mathbf{R}$ denote the matrix which components are $R(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, \cdots, n$. Then

$$\mathbf{Y}|\mathbf{x} \sim MVN_n(Z(\mathbf{x})\boldsymbol{\beta}, \Sigma).$$

We can see under the kriging model the random vector $\mathbf{Y}$ has multivariate normal distribution with mean vector determined by the part $Z(\mathbf{x})\boldsymbol{\beta}$ and variance-covariance matrix $\Sigma = \sigma^2\mathbf{R}$ where $\mathbf{R}$ satisfies a certain set of conditions in order for the $\Phi(\mathbf{x})$

process to be stationary. The linear mean part $Z(\mathbf{x})\boldsymbol{\beta}$ models the data trend and the variance-covariance part controls the smoothness and other properties of the model.

In order to understand the kriging model, we need to employ standard multivariate normal theory to obtain various useful properties of the model.

Suppose that

$$\mathbf{Y} = \begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \sim \mathrm{N}_{n+1}(\boldsymbol{\mu}, \Sigma), \ \boldsymbol{\mu} = \begin{pmatrix} \mu_0(\mathbf{x}_0) \\ \mu_1(\mathbf{x}_1) \\ \mu_2(\mathbf{x}_2) \\ \vdots \\ \mu_n(\mathbf{x}_n) \end{pmatrix} \tag{2.9}$$

where Σ is the variance-covariance matrix of vector $\mathbf{Y}$, the (i, j)-th element of the matrix Σ is

$$(\Sigma)_{ij} = \mathrm{Cov}(Y(\mathbf{x}_i), Y(\mathbf{x}_j)) \tag{2.10}$$

Partition $\mathbf{Y}$, $\boldsymbol{\mu}$ and Σ in the following way:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{pmatrix}, \ \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \tag{2.11}$$

Assume $\mathbf{Y}_1$ and $\boldsymbol{\mu}_1$ are vectors with n_1 elements, let $n_2 = n + 1 - n_1$, then $\mathbf{Y}_2$ and $\boldsymbol{\mu}_2$ are vectors with n_2 elements. From the theory of the multivariate normal distribution Arnold (1981) the following statements are valid:

- $\mathbf{Y}_i \sim \mathrm{N}_{n_i}(\boldsymbol{\mu}_i, \Sigma_{ii})$.
- $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are independent if and only if $\Sigma_{12} = 0$.
- Suppose $\Sigma_{jj} > 0$, then

$$\mathbf{Y}_i|\mathbf{Y}_j \sim \mathrm{N}_{n_i}(\boldsymbol{\mu}_i + \Sigma_{ij}\Sigma_{jj}^{-1}(\mathbf{Y}_j - \boldsymbol{\mu}_j), \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji}) \tag{2.12}$$

For the special case $n_1 = 1$, we have the observations $\mathbf{Y}_2 = (Y_1(\mathbf{x}_1), Y_2(\mathbf{x}_2), \cdots, Y_n(\mathbf{x}_n))'$, the conditional distribution of $Y_0(\mathbf{x}_0)$ given $\mathbf{Y}_2$ at $\mathbf{x} = \mathbf{x}_0$ is

$$\mathbf{Y}_1|\mathbf{Y}_2 = Y_0|\mathbf{Y}_2 \sim \mathrm{N}_1(\mu_0 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{Y}_2 - \boldsymbol{\mu}_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}). \tag{2.13}$$

Partition Σ

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \sigma^2 \begin{pmatrix} 1 & \mathbf{r}_0^T \\ \mathbf{r}_0 & \mathbf{R} \end{pmatrix} \tag{2.14}$$

where

$$\mathbf{R} = (R_{ij})_{n\times n},\ R_{ij} = corr(Y(\mathbf{x}_i), Y(\mathbf{x}_j)) \tag{2.15}$$

for $i = 1, \cdots, n$ and $j = 1, \cdots, n$. Then the conditional distribution of equation (2.13) can be written

$$Y_0|\mathbf{Y}_2 \sim \mathrm{N}_1(\mu_0 + \mathbf{r}_0^T\mathbf{R}^{-1}(\mathbf{Y}_2 - \boldsymbol{\mu}_2), \sigma^2(1 - \mathbf{r}_0^T\mathbf{R}^{-1}\mathbf{r}_0)). \tag{2.16}$$

Using the above multivariate normal theorems for the mean function which has the form of $\boldsymbol{\mu}(\mathbf{x}) = \mathbf{Z}(\mathbf{x})\beta$, where $\mathbf{Z}(\mathbf{x})$ is the basis matrix, it is straightforward to show that the best linear unbiased predictor (BLUP) for $Y(\mathbf{x}_0)$ at unobserved data point $\mathbf{x}_0$ given the known data point $\mathbf{Y} = (Y(\mathbf{x}_1), \cdots, Y(\mathbf{x}_n))'$

$$\hat{Y}(\mathbf{x}_0) = E(Y(\mathbf{x}_0)|\mathbf{Y}) = \mathbf{z}'(\mathbf{x}_0)\hat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x}_0)'\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{Z}\hat{\boldsymbol{\beta}}), \tag{2.17}$$

where $\mathbf{z}(\mathbf{x}_0) = (z_1(\mathbf{x}_0), \cdots, z_p(\mathbf{x}_0))'$, $\sigma^2(\mathbf{R})_{ij} = \sigma^2 R(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Cov}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))$, $\sigma^2\mathbf{r}(\mathbf{x}_0) = (\mathrm{Cov}(\Phi(\mathbf{x}_0), \Phi(\mathbf{x}_1)), \cdots, \mathrm{Cov}(\Phi(\mathbf{x}_0), \Phi(\mathbf{x}_n)))'$, and $\hat{\boldsymbol{\beta}}$ is the weighted least square estimator where $\hat{\boldsymbol{\beta}} = (\mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{R}^{-1}\mathbf{Y}$.

The variance for $\hat{Y}(\mathbf{x}_0)$ is

$$Var(\hat{Y}(\mathbf{x}_0)) = \sigma^2 - (\mathbf{z}'(\mathbf{x}_0), \mathbf{r}'(\mathbf{x}_0)) \begin{pmatrix} 0 & \mathbf{Z}' \\ \mathbf{Z} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{z}(\mathbf{x}_0) \\ \mathbf{r}(\mathbf{x}_0) \end{pmatrix} \tag{2.18}$$

The first part of (2.17) is the generalized least squares prediction at point $\mathbf{x}_0$ given the variance-covariance matrix $\Sigma = \sigma^2\mathbf{R}$. The second part (correction term) forces the regression line to pass through the observation data. The behavior of the "forcing" part is controlled by $\mathbf{R}$. The prediction at points $\mathbf{x}_i$ are exactly the corresponding observation and hence the variance of prediction at this point is zero. As the prediction point $\mathbf{x}_0$ moves away from the point $\mathbf{x}_i$, the second part of (2.17) goes to zero, which result in the generalized least squares estimator in the area far away from the observation.

To show this point, consider the correction term in (2.17) $\mathbf{r}(\mathbf{x}_0)'\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{Z}\hat{\boldsymbol{\beta}})$, which is a linear combination of the residuals $\mathbf{Y}-\mathbf{Z}\hat{\boldsymbol{\beta}}$ according to the model (2.7) with prediction point specific coefficients

$$\mathbf{r}(\mathbf{x}_0)'\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{Z}\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} c_i(\mathbf{x}_0)(\mathbf{Y}-\mathbf{Z}\hat{\boldsymbol{\beta}})_i$$

where $c_i(\mathbf{x}_0)$ is the ith element of vector $\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_0)$ and $(\mathbf{Y}-\mathbf{Z}\hat{\boldsymbol{\beta}})_i$ is the ith residual based on the fitted model.

To see why this correction term forces the predictor to interpolate the observation data, suppose that $\mathbf{x}_0 = \mathbf{x}_i$ for some fixed i, $1 \le i \le n$, then $\mathbf{z}(\mathbf{x}_0) = \mathbf{z}(\mathbf{x}_i)$, $\sigma^2\mathbf{r}(\mathbf{x}_0) = \sigma^2\mathbf{r}(\mathbf{x}_i) = (\mathrm{Cov}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_1)), \cdots, \mathrm{Cov}(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_n)))'$ which is the ith column of $\mathbf{R}$. Thus $\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_0) = (0, \cdots, 0, 1, 0, \cdots, 0)' = \mathbf{e}_i$, the ith unit vector, an n-component vector with ith element as unit and rest of elements are zeroes. This is because this product is the ith column of $\mathbf{R}^{-1}\mathbf{R} = \mathbf{I}_n$, the $n \times n$ identity matrix. Hence

$$\mathbf{r}(\mathbf{x}_0)'\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{Z}\hat{\boldsymbol{\beta}}) = \mathbf{e}_i'(\mathbf{Y}-\mathbf{Z}\hat{\boldsymbol{\beta}}) = Y_i - \mathbf{z}(\mathbf{x}_i)\hat{\boldsymbol{\beta}}$$

and hence

$$\hat{Y}(\mathbf{x}_0) = \hat{Y}(\mathbf{x}_1) = \mathbf{z}'(\mathbf{x}_i)\hat{\boldsymbol{\beta}} + Y_i - \mathbf{z}(\mathbf{x}_i)\hat{\boldsymbol{\beta}} = Y_i$$

when $\mathbf{x}_0 = \mathbf{x}_i$ which shows the prediction interpolate the observational data.

In the kriging model, the bias part $\Phi(\mathbf{x})$ is treated as a Stationary Gaussian Random Function with mean zero. *Stationary* means the distribution of $\Phi(\mathbf{x})$ does not depend on the location $\mathbf{x}$.

We can see that the distribution of $\Phi(\mathbf{x})$ solely depends on its variance-covariance matrix Σ or correlation matrix **R**. The selection of **R** plays a key role in the computer experiment modeling and in the predictive process. Koehler and Owen (1996) and Santner et al. (2004) summarize popular choices for **R** and their properties.

Basically, any positive definite matrix **R** can be used as a correlation matrix function, but for simplicity, people usually restrict this function to a stationary function such that $R(x_1, x_2) = R(x_1 - x_2)$ which only depends on the distance not on the actual location. A further restriction makes the correlation function only depend on the magnitude of the distance

$$R(x_1, x_2) = R(\|x_1 - x_2\|).$$

For multi-dimension problem, the correlation function will be simplified to

$$R(\mathbf{x}_1, \mathbf{x}_2) = \prod_{j=1}^{s} R_j(\|x_{1j} - x_{2j}\|).$$

Hence it's a product of univariate correlation functions and only univariate correlation function are of interested.

The choice of correlation function R is usually made based on considerations of smoothness, where smoothness reflects the number of times the random function Y is differentiable, There are several main types of univariate correlation functions. The cubic correlation family is given by

$$R(d) = 1 - \frac{3(1-\rho)}{2+\gamma} d^2 + \frac{(1-\rho)(1-\gamma)}{2+\gamma} |d|^3 \tag{2.19}$$

where $d \in [0,1]$ is the normalized distance, $\rho = corr(Y(0), Y(1)) \in [0,1]$ is the correlation between the end points observations, $\gamma = corr(Y'(0), Y'(1))$ is the correlation between the end point derivatives. Various version of this type can be obtained by using correlations between end point second derivatives and mixed situations in order to control the predictor's behavior on the boundaries. The prediction model in one dimension based on this type of correlation functions is a cubic spline interpolator.

The exponential correlation function has the following form:

$$R(d) = \exp(-\theta|d|) \tag{2.20}$$

for the normalized distance d and $\theta \in (0, \infty)$. Processes with this type of correlation functions are called Ornstein-Uhlenbeck processes. The predictors with this kind of correlation functions are non-differentiable.

The above exponential correlation function can be generalized into the following form (Sacks et al. (1989b)):

$$R(d) = \exp(-\theta|d|^q) \tag{2.21}$$

where $0 < q \leq 2$ and $\theta \in (0, \infty)$. When $q = 1$, this correlation function turns into the exponential correlation function. As q increases, this function will result in a smoother predictor. But as long as $q < 2$, these processes are not mean square differentiable. The so-called Gaussian correlation function is the case when $q = 2$. The corresponding processes are infinitely mean square differentiable.

Note that the cubic correlation function is twice differentiable, the exponential and Gaussian family are zero and infinitely many mean square differentiable. The Matern correlation function (Matern (1947); Yaglom (1987); Stein (1989)) is a more flexible one through which the degree of smoothness can be controlled. It has the following form:

$$R(d) = \frac{(\theta|d|)^\mu}{\Gamma(\mu)2^{\mu-1}} K_\mu(\theta|d|) \tag{2.22}$$

where $\theta \in (0, \infty)$ and $\mu \in (-1, \infty)$. $K_\mu(.)$ is the modified Bessel function of order μ. When $\mu > m$, the associated process will be m times differentiable. The degree of differentiability can be controlled by parameter μ and θ controls the range of the correlations.

The distribution of the random vector **Y** depends solely on the coefficient vector $\boldsymbol{\beta}$ and the variance-covariance matrix Σ in the kriging model. Given Σ, the general least squares estimator is the best linear unbiased estimator. But in practice, the variance-covariance matrix is unknown to us. Except for the unknown parameter σ^2, in the various correlation functions aforementioned, there are either one or two unknown parameters which will fully determine the corresponding correlation structure.

There exist several approach to estimating all these unknown parameters. Lehman (2002) compared several methods to estimate the unknown parameters in these models and recommended the maximum likelihood estimator (MLE) and restricted maximum likelihood estimator (ReMLE). It is a widely known fact that MLE is a biased estimator for σ^2 while $ReMLE$ is an unbiased estimator for σ^2. But ReMLE can not estimate the coefficient vector $\boldsymbol{\beta}$ (this is where the "restricted" comes from).

Both methods attempt to minimize likelihood related quantities. MLE works on the original likelihood function and the MLE for $\boldsymbol{\beta}$ is the general least squares estimator. The MLE for σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}(\mathbf{Y} - \mathbf{Z}\hat{\boldsymbol{\beta}})'\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{Z}\hat{\boldsymbol{\beta}}). \tag{2.23}$$

Both $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are functions of **R**, which is either a function of parameter θ or θ and μ.

Unfortunately, the normal equations for θ and other unknown parameters in correlation functions are generally complicated and do not yield any analytic solution. So numerical algorithm can be used to iteratively get the MLEs for $\boldsymbol{\beta}$, σ^2 and the parameters in correlation functions simultaneously. See Mardia and Marshall (1984) for an overview of the MLE procedure. The ReMLE is very similar to MLE except that it deals with only the part of likelihood function which is related to the unknown parameters in the correlation function. It also needs an iterative approach to get the estimator. When the sample size is small, the likelihood function becomes flat around the maximizer which leads to the resulting kriging interpolation behaving erratically. Li and Sudjianto (2005) proposed a penalized likelihood approach for the Gaussian kriging model to deal with such an issue.

2.4 Local Polynomial Regression

The polynomial and spline regressions are examples of global approximations in nonparametric smoothing methods. As we have seen, the linear regression and polynomial regression will have a large bias because the degree of the polynomial regression can not be controlled locally and continuously. At the same time, individual observations can have a big influence on remote parts of the curve in a polynomial model. Instead of considering the coefficients of the predictors as constants, local polynomial regression makes these coefficients vary according to the predictor values and are determined only by the data which are near the current location. This is where the word 'local' comes from. Hence it aims to relax assumptions on the form of regression function to let data search for a suitable form. It can be used in other statistical problems such as goodness of fit tests. It can deal with functional data and longitudinal data. It is very flexible, can reduce possible model bias, and can explore fine structure relationship easily.

There are two main kinds of local modeling techniques, kernel and local polynomial(including local linear) regression. One found that the smoothing splines are asymptotically equivalent to kernel regression. Both smoothing splines and local linear regression are the most commonly used smoothing methods though the penalized spline is the easiest to implement.

Fan and Gijbels (1996) systematically summarized the theoretical properties of local polynomial regression. It is proven to be the best linear estimator, the best boundary correction method, and nearly the best among nonlinear estimators. Although there is a lot of literature on this method, it has not been a popular approach in modeling computer experiments. Hϕst (1999) used the local polynomial regression with kriging to model computer experiments. Tu and Jones (2003) present some studies on how to use local polynomial regression to fit computer experiment data. These works show the possibility of local polynomial regression for computer experiments.

2.4.1 Kernel regression

Consider a one-dimensional non-parametric regression model. Suppose (x_i, y_i), $i = 1, \cdots, n$ is a random sample from the following model:

$$Y = m(X) + \epsilon \tag{2.24}$$

where $E(\epsilon|X = x) = 0$ and $Var(\epsilon|X = x) = \sigma^2(x)$. $m(x) = E(Y|X = x)$ is called the regression function. The goal of non-parametric regression is to estimate $m(x)$ without assuming any parametric form on the regression function. A naive way for the estimating regression function is to use a local average. An improved version of this is the locally weighted average

$$\hat{m}(x) = \frac{\sum_{i=1}^{n} w_i(x) y_i}{\sum_{i=1}^{n} w_i(x)}, \tag{2.25}$$

where $w_i(x), i = 1, \cdots, n$ are weights. This estimator can also be interpreted as a solution of a weighted least square problem

$$\min_{m(x)} \sum_{i=1}^{n} (y_i - m(x))^2 w_i(x). \tag{2.26}$$

A kernel function is used to obtain the weight function $w_i(x)$. Let $K(x)$ be a kernel function which satisfies $\int K(x)dx = 1$ and h is a positive number called the bandwidth or smoothing parameter. Take the weight function $w_i(x)$ to be $h^{-1}K((x_i -$

$x)/h)$, denoted by $K_h(x_i - x)$. We want to use more information from the observations which are near the current location x so that at point x (or at origin of function K_h), the kernel function should have the highest value, and we want the observations which are far from the point x to have smaller influence on this point. Thus, when the magnitude of $x_i - x$ is large, the kernel function should be very small. Then the estimator (2.25) becomes

$$\hat{m}_h(x) = \frac{\sum_{i=1}^n K_h(x_i - x)y_i}{\sum_{i=1}^n K_h(x_i - x)}, \tag{2.27}$$

which is referred to as NW-kernel estimator and was proposed independently by Nadaraya (1964) and Watson (1963).

Another kind of kernel smoothing estimator is called *GM* estimator (Gasser and Muller) in the following form

$$\hat{m}_h(x) = \sum_{i=1}^n \int_{s_{i-1}}^{s_i} K_h(u - x) du y_i, \tag{2.28}$$

where $s_i = (x_{(i)} + x_{(i+1)})/2$, $x_{(0)} = -\infty$ and $x_{(n+1)} = +\infty$ and $x_{(i)}$ is the ith order statistics.

2.4.2 Local polynomial(including local linear) regression

The NW-kernel estimator is actually a weighted least square estimator using a local constant approximation. It can be directly extended to local polynomial regression. Suppose $m(x)$ is smooth, applying a Taylor expansion for $m(x)$ in a neighborhood of x (for example, for x_i close to x)

$$m(x_i) \approx \sum_{j=0}^p \frac{m^{(j)}(x)}{j!}(x_i - x)^j = \sum_{j=0}^p \beta_j (x_i - x)^j = \mathbf{x}_i' \boldsymbol{\beta}, \tag{2.29}$$

where $\mathbf{x}_i = (1, (x_i - x), \cdots, (x_i - x)^p)'$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)$, $\beta_j = \frac{m^{(j)}(x)}{j!}$. Intuitively, data points close to x will have more influence about $m(x)$. This suggests

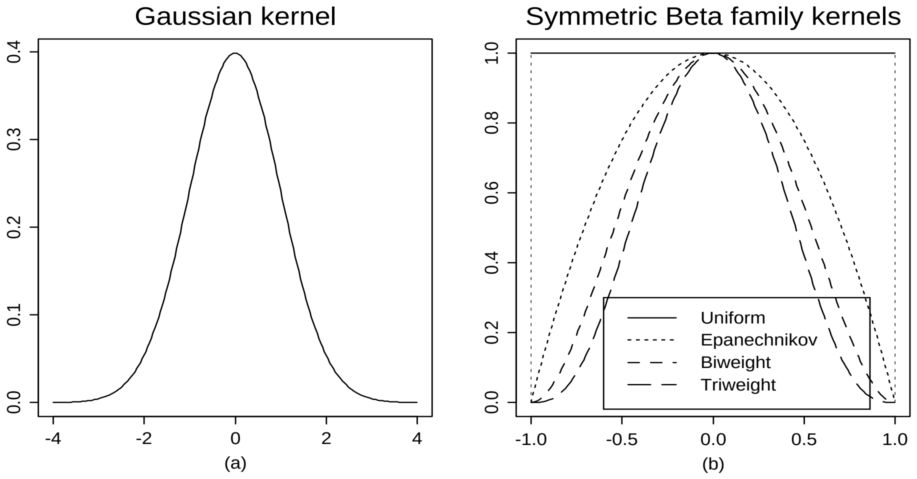


Figure 2.1. Kernel functions

using a locally weighted polynomial regression

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i \boldsymbol{\beta})^2 K_h(x_i - x). \tag{2.30}$$

Denote $W = diag(K_h(x_1 - x), \cdots, K_h(x_n - x))$, and the design matrix associated with $\mathbf{x}_i, i = 1, n$ as **X**, then the solution to locally weighted least squares (2.30) is

$$\hat{\boldsymbol{\beta}}(x) = (\mathbf{X}' W \mathbf{X})^{-1} \mathbf{X}' W \mathbf{Y}, \tag{2.31}$$

where $\mathbf{Y} = (y_1, \cdots, y_n)'$. The estimator for the regression function $m(x)$ is

$$\hat{m}(x) = \hat{\beta}_0(x).$$

Furthermore, an estimator for the ν-th order derivative of $m(x)$ at x is

$$\hat{m}_\nu(x) = \nu! \hat{\beta}_\nu(x).$$

The choice of kernel function is not sensitive to the estimation of $m(x)$ (Marron and Nolan (1988)). The symmetric kernel function has many advantages over asymmetric functions. The most significant one is that it will have better asymptotic properties. The most commonly used kernel function is the Gaussian kernel function

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2). \tag{2.32}$$

Another popular choice for the kernel function is the symmetric beta family

$$K(x) = \frac{1}{Beta(1/2, \gamma + 1)}(1 - x^2)_+^\gamma, \quad \gamma = 0, 1, \cdots \tag{2.33}$$

The support of this kernel function is [-1,1] and $Beta(.,.)$ is the beta function. The corresponding kernel function when $\gamma = 0, 1, 2$ and 3 are the uniform, the Epanechnikov, the biweight and the triweight kernel function (Fan and Gijebls, 1996). The optimal kernel function is proved to be the Epanechnikov kernel $K_{opt} = 0.75(1 - x^2)_+$ but other symmetric kernel functions have comparable performance. Figure 2.1 depicts the commonly used kernel functions.

The smoothing parameter h controls the smoothness of regression function. The choice of the bandwidth is of crucial importance. If h is too large, then the smoothing result will miss the fine structure of the data. On the other hand, if h is too small, then the noise will show up too prominently. Jones et al. (1996a) and Jones et al. (1996b) gave a systematic review on bandwidth selection. In theory, one can obtain the optimal bandwidth by minimizing the *MISE* which is a function of h. In practice, data driven methods are utilized to choose the bandwidth by visualizing the resulting estimated regression function.

The choice for the order of the local polynomial regression p is highly associated with ν, the number of derivatives that the mean function $m(x)$ attempts to approximate. Asymptotic results show that in order to obtain higher order accuracy, we need to choose p such that $p = \nu + 1$ or $p = \nu + 3$. So if estimating $m(x)$ is our purpose, then p needs to be 1 or 3. Hence we need to use local linear or local cubic regression to estimate the mean function.

Since in many applications more than 50% of data points are near the boundary, the boundary effect of the estimator is a very important problem. Most nonparametric smoothers such as NW and GM estimators behave poorly near the boundary.

Local polynomial regression performs well on this point. When $p - \nu$ is odd, the bias and variance will change continuously from the interior to the boundary and hence will best eliminate the boundary effect.

The asymptotic properties of the estimates using symmetric kernels is summarized as follows:

Table 2.1. *Leading terms in the asymptotic biases and variances*
$b_n = \frac{1}{2}\int_{-\infty}^{+\infty} u^2 K(u) du h^2$, $V_n = \frac{\sigma^2(x)}{f(x)nh}\int_{-\infty}^{+\infty} K^2(u) du$

Method	Bias	Variance
NW estimator	$m''(x) + \frac{2m'(x)f'(x)}{f(x)} b_n$	V_n
GM estimator	$m''(x) b_n$	$1.5V_n$
Local linear	$m''(x) b_n$	V_n

Although the local polynomial regression was originally proposed for nonparametric regression, it can be applied directly for modeling computer experiments where there is no random error ϵ.

2.5 Functional Linear Model

In general, data collected over an interval of some index with an intensive sampling rate are referred to as functional data. For example, data were collected every 20 seconds over a period of three days; and image data can be viewed as functional data collected over a 2-dimensional interval when a set of image samples are available.

Unlike longitudinal data which are sparse and collected irregularly, functional data are more dense and regular; hence, we can treat the response as a smooth function. With the rapid development of computer technology and devices for collecting data, people can easily collect and store functional data. Functional data analysis is becoming popular in various research fields. Many techniques have been introduced for the analysis of functional data which allows the exploration of variability in sample of curves and allows for the research of common structures among the curves or groups of them. A very interesting application consists of the possibility of constructing regression models through which we describe the relation between a real variable and an explanatory variable having a functional nature. Such a model is termed a functional regression model.

Various interesting examples of functional linear models can be found in literature. Ramsay and Silverman (1997) present many interesting examples of functional data and introduce various statistical models to fit the functional data. Hastie and Mallows (1993) proposed some application to quantitative chemistry, and Marx and Eilers (1996) illustrated a phonemes classification by means of the log-spectra of a sequence of spoken syllables. Cardot et al. (1999) proposed a forecasting model in order to explain winter wheat yield as a linear function of the duration of the crop and climatic variations. Ferraty and Vieu (2002) illustrated the prevision of fat content in some meat samples from the respective spectrometric curves. Li et al. (2005) studied the computer output of the functional motion error of the valvetrain system under varies design cases. Zhang (2005) studied modeling computer experiments in detail with functional responses using nonparametric and semi-parametric models.

The functional linear model originated from varying coefficient models which were introduced in Hastie and Tibshrani (1993). In this model, suppose we have a random variable Y whose distribution depends on η, and we also have predictors $X_1, \cdots, X_p$ and $R_1, \cdots, R_p$, then the varying coefficient model has the following form:

$$\eta = \beta_0 + X_1\beta_1(R_1) + \cdots + X_p\beta_p(R_p), \tag{2.34}$$

which says the coefficients $\beta_1, \cdots, \beta_p$ are functions of variables $R_1, \cdots, R_p$, and $\eta = g(\mu)$ is the link function in the generalized linear models where $\mu = E(Y|\mathbf{X})$.

For a special case of this model, the R_js will often be the same variable, such as time, angle, degree or distance. Suppose for predictors $X_1, \cdots, X_p$, we have dense measurements Y each over times $t = (t_1, \cdots, t_m)$. Then we can model this as

$$\eta(t) = \beta_0(t) + X_1\beta_1(t) + \cdots + X_p\beta_1(t) \tag{2.35}$$

This model is referred to time-varying coefficient model. Furthermore, in (2.35), if for all observations, the measurements happens at the same time $t_j, j = 1, \cdots, m$, then this model is called a functional linear model. From this, we can see that for a given time t, functional linear models are nothing more than merely an ordinary linear model. As t changes, the relationship between the response Y with the predictors X_i changes according to the relationship between t and β_js. From (2.35), we can see that the functional linear model relaxes the assumption for the linear model, allowing the variation of the coefficient with some variable hence increasing the flexibility of the linear model which will greatly reduce the model bias.

Faraway (1997) applied smoothing splines to calculate the function $Y_i(t)$ in the varying-coefficient model for each observation in the case where the measurements occur at different time location. Then he used least squares to get the estimation for β_j at each time location. Hoover et al. (1998) proposed two nonparametric estimators for the smoothed functions $\beta_j(t)$. One is by an orthogonal B-spline basis; another is by local polynomial regression. They also proposed a cross-validation criterion for selecting the corresponding smoothing parameter. Wu et al. (1998) studied the asymptotic properties of the local polynomial regression estimator for $\beta_j(t)$. These asymptotic results are used to construct the pointwise and simultaneous confidence region for $\beta_j(t)$. In these approaches, the estimators for smooth functions $\beta_j(t)$ are one-step estimators. These estimating procedures have several drawbacks, such as the implicit assumption of all the functions $\beta_j(t), j = 0, \cdots, p$ possessing the same degree of smoothness. In addition, the computation is not efficient, perhaps undersmoothing some of the underlying coefficient functions when these functions have different degrees of smoothness.

Fan and Zhang (1999) addressed the first drawback of the one-step estimator and proposed a two-step estimation procedure. They take the one-step estimation result as their first step result, then treat the predictors with different degrees of smoothness, and augment the optimality of the estimator in their second step estimation.

Another two-step estimation procedure which was proposed by Fan and Zhang (2000) overcomes the drawback of the inefficiency and computational expense of the one-step estimator. For simplicity of description, assume that all the data is observed at the same time and at dense points $t_j, j = 1, \cdots, m$. This can be achieved by binning the functional data with respect to observed times t_{ij} if necessary. They suggested that in the first step, we estimate $\beta(t_j)$ by linear regression using n data points collected at time t_j. They use the local polynomial regression to smooth $(t_j, \beta(t_j)), j = 1, \cdots, m$ componentwisely, having estimated $\beta(t)$ over $t_1, \cdots, t_m$ in the second step. This procedure can be easily implemented and allows different coefficients to have different degrees of smoothness. Li, Sudjianto and Zhang (2005) applied the two-step estimation procedure to a functional linear model with time-invariant covariate and use penalized splines in the second step.

2.6 Partial Functional Linear Model

In most situations, we will find that the smoothed functional coefficients have different levels of dependency on the index variable t. Some of them strongly depend on t while others depend on it weakly. Partial functional linear models deal with this situation and incorporate these two kinds of coefficients as:

$$y = \mathbf{x}^T \boldsymbol{\alpha}(t) + \mathbf{z}^T \boldsymbol{\beta} + \epsilon \tag{2.36}$$

where both **x** and **z** are covariate vectors, $\boldsymbol{\alpha}(t)$ consists of p unknown smooth functions, $\boldsymbol{\beta}$ is a q-dimensional unknown parameter vector, and $E(\epsilon|\mathbf{x}, \mathbf{z}) = 0$.

Partially functional linear models are a natural extension of partially linear models which are defined as:

$$y = \alpha(t) + \mathbf{x}^T \boldsymbol{\beta} + \epsilon, \tag{2.37}$$

where $\alpha(t)$ is a smooth function of t, $\boldsymbol{\beta}$ is an unknown regression coefficient vector and ϵ is a random error with $E(\epsilon|t, x) = 0$. Zeger and Diggle (1994) suggested using a backfitting algorithm to find an estimate for $\alpha(t)$ and $\boldsymbol{\beta}$. Specifically, starting with an initial value of $\boldsymbol{\beta}$, denoted by $\boldsymbol{\beta}^{(0)}$, we smooth the residual $y_i(t_{ij}) - \mathbf{x}_i^T \boldsymbol{\beta}^{(0)}$ over t_{ij} to estimate $\alpha(t)$. Having an estimate for $\alpha(t)$, denoted by $\hat{\alpha}(t)$, we conduct linear regression of $y_i(t_{ij}) - \hat{\alpha}(t_{ij})$ on $\mathbf{x}_i(t_{ij})$. Iterate this procedure until it converges. This is basically the same as the Gauss-Seidal algorithm to compute the profile least squares estimate. Moyeed and Diggle (1994) proposed an improved version of the backfitting algorithm based on a partial residual approach.

Lin and Ying (2001) introduced the counting process technique to the estimation scheme for the partially linear model. Fan and Li (2004) proposed two estimators: a difference-based estimator(DBE) and a profile least squares estimator. In DBE, rewrite the observed data $\{(t_{ij}, \mathbf{x}(t_{ij}), y(t_{ij})), j = 1, \cdots, J_i, i = 1, \cdots, n\}$ as $\{(t_i, \mathbf{x}(t_i), y(t_i)), i = 1, \cdots, n^*\}$ where $n^* = \sum_{i=1}^n J_i$. Then model (2.37) turns into:

$$y_i = \alpha(t_i) + \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i \tag{2.38}$$

Take the difference between $i + 1$ and i, we have:

$$y_{i+1} - y_i = \alpha(t_{i+1}) - \alpha(t_i) + \boldsymbol{\beta}^T (\mathbf{x}_{i+1} - \mathbf{x}_i) + e_i \tag{2.39}$$

where $e_i = \epsilon_{i+1} - \epsilon_i$. Under some mild conditions, the spacing between t_i and t_{i+1} is of order $O(1/n)$. Hence, the term $\alpha(t_{i+1}) - \alpha(t_i)$ is negligible. The least-squares approach can be employed to estimate the parameter . The method can be further improved by fitting the following linear model:

$$y_{i+1} - y_i = \alpha_0 + \alpha_1(t_{i+1} - t_i) + \boldsymbol{\beta}^T(\mathbf{x}_{i+1} - \mathbf{x}_i) + e_i \tag{2.40}$$

Again, a least square approach can be utilized to get the estimation for $\boldsymbol{\beta}$.

In the profile least squares approach, Fan and Li (2004) rewrite model (2.37) as:

$$y^*(t) = y(t) - \mathbf{x}^T\boldsymbol{\beta} = \alpha(t) + \epsilon \tag{2.41}$$

and use local linear regression to get $\hat{\alpha}(t)$ the approximation for $\alpha(t)$. It is well known that the local linear fit is linear in $y^*(t)$ (Fan and Gijbels, 1996). Thus, the estimate of $\alpha(t)$ is linear in $y(t) - \mathbf{x}^T\boldsymbol{\beta}$. Hence, the estimate for the vector $\boldsymbol{\alpha}$ can be expressed as $\hat{\boldsymbol{\alpha}} = \mathbf{S}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$. The matrix $\mathbf{S}$ is usually called a smoothing matrix of the local linear smoother. It depends only on the observation times $\{t_{ij}\}$ and the amount of smoothing h. Substituting b into (3.13), we get:

$$(\mathbf{I} - \mathbf{S})\mathbf{y} = (\mathbf{I} - \mathbf{S})\mathbf{X}\boldsymbol{\beta} + \epsilon \tag{2.42}$$

where $\mathbf{I}$ is the identity matrix with order $n^* = \sum_{i=1}^n J_i$. The weighted least square estimation of $\boldsymbol{\beta}$ for above model is:

$$\hat{\boldsymbol{\beta}} = \{\mathbf{X}^T(\mathbf{I} - \mathbf{S})^T\mathbf{W}(\mathbf{I} - \mathbf{S})\mathbf{X}\}^{-1}\mathbf{X}^T(\mathbf{I} - \mathbf{S})^T\mathbf{W}(\mathbf{I} - \mathbf{S})\mathbf{y} \tag{2.43}$$

where $\mathbf{W}$ is the weight matrix in the general least-squares, which can incorporate the within subject correlation. The above estimator is called the profile least-squares estimator. The profile least-squares estimator for the nonparametric component is simply $\alpha(t : \hat{\boldsymbol{\beta}})$.

Martinussen and Scheike (1999) proposed an estimation procedure for the model (2.36) using the notion of a counting process. Sun and Wu (2005) extended the estimation procedure of Lin and Ying for partially linear models to model (2.36). Fan, Huang and Li (2005) extended the profile least squares approach for model (2.36), and further proposed semi-parametric modeling strategy for the covariance function of random error process $\epsilon(t)$.

Chapter 3

Modeling Computer Experiments with Multiple Responses

The goal of this chapter is to develop models and procedures for computer experiments with sparse functional response. In this chapter, we view the sparse functional response as a multiple response. It has been commonly known that one may collect multiple responses from a physical or computer experiment. However, to the best of our knowledge, there is little work to model computer experiments with multiple responses. In this chapter, we propose a modeling procedure for such computer experiments by using a multivariate kriging model, a natural extension to the ordinary kriging model. We further extend functional ANOVA of single response to multiple responses and apply it to analyzing the effect of each design variable. The proposed methodology is demonstrated by an analysis of Example 1.1 in the introduction chapter, whose data is collected from a case study concerned with the design of the engine structure to minimize the radiated noise.

The structure for this chapter is organized as follows. In Section 3.1, we review the theory of matrix normal distributions. Section 3.2 presents a Gaussian Kriging model for computer experiments with multiple responses. Section 3.3 gives an estimation procedure for the Gaussian Kriging model. In a practical implementation, one needs a numerical method to estimate the parameters in the Gaussian Kriging model, and the first and second derivatives of the matrix. We also include these derivatives in Section 3.3. We introduce the multivariate functional ANOVA in Section 3.4 in order to study the effect of each variables. We conduct and describe some empirical justification by Monte Carlo simulation in Section 3.5. Finally, we conduct a detailed analysis of Example 1.1 from Chapter 1 in Section 3.6.

3.1 Theory of Matrix Normal Distribution

Suppose **Z** is a $(n+1) \times (s+1)$ random matrix with the (i, j)-th element of matrix **Z** having a standard normal distribution

$$Z_{ij} \sim \mathrm{N}_1(0, 1), \tag{3.1}$$

where $i = 0, 1, \cdots, n$ and $j = 0, 1, \cdots, s$. Let $\Sigma \geq 0$ and $\Psi \geq 0$ be a $(n+1) \times (n+1)$ and a $(s+1) \times (s+1)$ symmetric matrix respectively and let $\boldsymbol{\mu}$ be $(n+1) \times (s+1)$ matrix. Define

$$\mathbf{Y} = \Sigma^{\frac{1}{2}} \mathbf{Z} \Psi^{\frac{1}{2}} + \boldsymbol{\mu}, \tag{3.2}$$

then **Y** has a matrix normal distribution with mean matrix $\boldsymbol{\mu}$ and covariance matrices Σ and Ψ. Write

$$\mathbf{Y} \sim \mathrm{N}_{n+1,s+1}(\boldsymbol{\mu}, \Sigma, \Psi). \tag{3.3}$$

Express **Y** in column format to be $\mathbf{Y} = (\mathbf{Y}_0, \mathbf{Y}_1, \cdots, \mathbf{Y}_s)$. Define

$$vec(\mathbf{Y}) = \left(\mathbf{Y}_0', \mathbf{Y}_1', \mathbf{Y}_2', \cdots, \mathbf{Y}_s'\right), \tag{3.4}$$

and the Kronecker product of Σ and Ψ to be

$$\Sigma \otimes \Psi = \begin{pmatrix} \Sigma_{00}\Psi & \cdots & \Sigma_{0n}\Psi \\ \Sigma_{10}\Psi & \cdots & \Sigma_{1n}\Psi \\ \vdots & \ddots & \vdots \\ \Sigma_{n0}\Psi & \cdots & \Sigma_{nn}\Psi \end{pmatrix} \tag{3.5}$$

Then the distribution of random matrix **Y** can be written as the usual matrix normal

$$vec(\mathbf{Y}) \sim \mathrm{N}_{(n+1)(s+1)}(vec(\boldsymbol{\mu}), \Sigma \otimes \Psi) \tag{3.6}$$

It is worth noting that Σ is the variance-covariance matrix between the rows of the random matrix $\mathbf{Y}$, while Ψ is the variance-covariance matrix between the columns of the random matrix $\mathbf{Y}$. When the number of columns is one ($s = 0$), Ψ becomes a positive number, denoted by σ^2, from the definition (3.2), we can see in this special case ($s = 0$),

$$\mathbf{Y} \sim \mathrm{N}_n(\boldsymbol{\mu}, \sigma^2\Sigma). \tag{3.7}$$

One may partition the observation matrix $\mathbf{Y}$ into two parts in two different ways. The first way is to partition it by columns, the second way is by rows. Partition $\mathbf{Y}$, $\boldsymbol{\mu}$ and Ψ to be

$$\begin{aligned} \mathbf{Y} &= (\mathbf{Y}_0, \mathbf{Y}_1, \cdots, \mathbf{Y}_s) = (\tilde{\boldsymbol{Y}}_1, \tilde{\boldsymbol{Y}}_2), \\ \boldsymbol{\mu} &= (\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_s) = (\tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\mu}}_2), \\ \Psi &= \begin{pmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22}, \end{pmatrix} \end{aligned} \tag{3.8}$$

where $\tilde{\boldsymbol{Y}}_1$ and $\tilde{\boldsymbol{\mu}}_1$ are $(n+1) \times s_1$ matrix, let $s_2 = s + 1 - s_1$, then $\tilde{\boldsymbol{Y}}_2$ and $\tilde{\boldsymbol{\mu}}_2$ are $(n+1) \times s_2$ matrix. As natural extensions of the multivariate normal distribution, the following statements hold:

(1) $\tilde{\boldsymbol{Y}}_i \sim \mathrm{N}_{n,s_i}(\tilde{\boldsymbol{\mu}}_i, \Sigma, \Psi_{ii})$

(2) $\tilde{\boldsymbol{Y}}_1$ and $\tilde{\boldsymbol{Y}}_2$ are independent iff $\Psi_{12} = 0$

(3) Suppose $\Psi_{jj} > 0$, then

$$\tilde{\boldsymbol{Y}}_i | \tilde{\boldsymbol{Y}}_j \sim \mathrm{N}_{n,s_i}(\tilde{\boldsymbol{\mu}}_i + (\tilde{\boldsymbol{Y}}_j - \tilde{\boldsymbol{\mu}}_j)\Psi_{jj}^{-1}\Psi_{ij}, \Sigma, \Psi_{ii} - \Psi_{ij}\Psi_{jj}^{-1}\Psi_{ji}) \tag{3.9}$$

<u>Proof</u> Note that

$$vec(\mathbf{Y}) = \begin{pmatrix} vec(\tilde{\boldsymbol{Y}}_1) \\ vec(\tilde{\boldsymbol{Y}}_2) \end{pmatrix} \sim \mathrm{N}_{(n+1)(s+1)}(\begin{pmatrix} vec(\tilde{\boldsymbol{\mu}}_1) \\ vec(\tilde{\boldsymbol{\mu}}_2) \end{pmatrix}, \begin{pmatrix} \Sigma \otimes \Psi_{11} & \Sigma \otimes \Psi_{12} \\ \Sigma \otimes \Psi_{21} & \Sigma \otimes \Psi_{22} \end{pmatrix}) \tag{3.10}$$

From the vector normal distribution result (2.12), we have

$$\begin{aligned}
& vec(\tilde{\boldsymbol{Y}}_i)|vec(\tilde{\boldsymbol{Y}}_j) \\
\sim \quad & \mathrm{N}_{(n+1)s_i}(vec(\tilde{\boldsymbol{\mu}}_i) + (\Sigma \otimes \Psi_{ij})(\Sigma \otimes \Psi_{jj})^{-1}(vec(\tilde{\boldsymbol{Y}}_j) - vec(\tilde{\boldsymbol{\mu}}_j)), \\
& \Sigma \otimes \Psi_{ii} - (\Sigma \otimes \Psi_{ij})(\Sigma \otimes \Psi_{jj})^{-1}(\Sigma \otimes \Psi_{ji})) \\
= \quad & \mathrm{N}_{(n+1)s_i}(vec(\tilde{\boldsymbol{\mu}}_i) + [\mathbf{I}_{(n+1)(n+1)} \otimes (\Psi_{ij}\Psi_{jj})^{-1}]vec(\tilde{\boldsymbol{Y}}_j - \tilde{\boldsymbol{\mu}}_j), \\
& \Sigma \otimes \Psi_{ii} - \Sigma \otimes (\Psi_{ij}\Psi_{jj}^{-1}\Psi_{ji})) \\
= \quad & \mathrm{N}_{(n+1)s_i}(vec(\tilde{\boldsymbol{\mu}}_i + (\tilde{\boldsymbol{Y}}_j - \tilde{\boldsymbol{\mu}}_j)\Psi_{jj}^{-1}\Psi_{ji}), \Sigma \otimes (\Psi_{ii} - \Psi_{ij}\Psi_{jj}^{-1}\Psi_{ji})) \qquad (3.11)
\end{aligned}$$

Hence,

$$\tilde{\boldsymbol{Y}}_i|\tilde{\boldsymbol{Y}}_j \sim \mathrm{N}_{n+1,s_i}(\tilde{\boldsymbol{\mu}}_i + (\tilde{\boldsymbol{Y}}_j - \tilde{\boldsymbol{\mu}}_j)\Psi_{jj}^{-1}\Psi_{ji}, \Sigma, \Psi_{ii} - \Psi_{ij}\Psi_{jj}^{-1}\Psi_{ji}) \qquad (3.12)$$

■

We next partition $\mathbf{Y}$, $\boldsymbol{\mu}$ and Σ to be

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_0 \\ \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_n \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{Y}}_1 \\ \tilde{\boldsymbol{Y}}_2 \end{pmatrix},$$

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_0 \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \vdots \\ \boldsymbol{\mu}_n \end{pmatrix} = \begin{pmatrix} \tilde{\boldsymbol{\mu}}_1 \\ \tilde{\boldsymbol{\mu}}_2 \end{pmatrix}, \ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \qquad (3.13)$$

where $\tilde{\boldsymbol{Y}}_1$ and $\tilde{\boldsymbol{\mu}}_1$ are $n_1 \times (s+1)$ matrix, let $n_2 = n + 1 - n_1$, then $\tilde{\boldsymbol{Y}}_2$ and $\tilde{\boldsymbol{\mu}}_2$ are $n_2 \times (s+1)$ matrix. The following statements are direct extension of multivariate normal distribution.

(1) $\tilde{\boldsymbol{Y}}_i \sim \mathrm{N}_{n_i,s}(\tilde{\boldsymbol{\mu}}_i, \Sigma_{ii}, \Psi)$

(2) $\tilde{\boldsymbol{Y}}_1$ and $\tilde{\boldsymbol{Y}}_2$ are independent iff $\Sigma_{12} = 0$

(3) Suppose $\Sigma_{jj} > 0$, then

$$\tilde{\boldsymbol{Y}}_i|\tilde{\boldsymbol{Y}}_j \sim \mathrm{N}_{n_i,s}(\tilde{\boldsymbol{\mu}}_i + \Sigma_{ij}\Sigma_{jj}^{-1}(\tilde{\boldsymbol{Y}}_j - \tilde{\boldsymbol{\mu}}_j), \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji}, \Psi) \tag{3.14}$$

<u>**Proof**</u> It is known that if $\mathbf{Y} \sim \mathrm{N}_{n+1,s+1}(\boldsymbol{\mu}, \Sigma, \Psi)$, then

$$\mathbf{Y}^T \sim \mathrm{N}_{s+1,n+1}(\boldsymbol{\mu}^T, \Psi, \Sigma) \tag{3.15}$$

and

$$\mathbf{Y}^T = (\tilde{\boldsymbol{Y}}_1^T, \tilde{\boldsymbol{Y}}_2^T) \sim \mathrm{N}_{s+1,n+1}((\tilde{\boldsymbol{\mu}}_1^T, \tilde{\boldsymbol{\mu}}_2^T), \Psi, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}). \tag{3.16}$$

From the results of partition $\mathbf{Y}$ by columns, it follows that

$$\begin{aligned} & vec(\tilde{\boldsymbol{Y}}_i^T)|vec(\tilde{\boldsymbol{Y}}_j^T) \\ \sim \quad & \mathrm{N}_{(s+1)n_i}(vec(\tilde{\boldsymbol{\mu}}_i^T + (\tilde{\boldsymbol{Y}}_j^T - \tilde{\boldsymbol{\mu}}_j^T)\Sigma_{jj}^{-1}\Sigma_{ji}), \Psi \otimes (\Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji})). \end{aligned} \tag{3.17}$$

Hence,

$$\tilde{\boldsymbol{Y}}_i^T|\tilde{\boldsymbol{Y}}_j^T \sim \mathrm{N}_{s+1,n_i}(\tilde{\boldsymbol{\mu}}_i^T + (\tilde{\boldsymbol{Y}}_j^T - \tilde{\boldsymbol{\mu}}_j^T)\Sigma_{jj}^{-1}\Sigma_{ji}, \Psi, \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji}), \tag{3.18}$$

and finally,

$$\tilde{\boldsymbol{Y}}_i|\tilde{\boldsymbol{Y}}_j \sim \mathrm{N}_{n_i,s+1}(\tilde{\boldsymbol{\mu}}_i + \Sigma_{ij}\Sigma_{jj}^{-1}(\tilde{\boldsymbol{Y}}_j - \tilde{\boldsymbol{\mu}}_j), \Sigma_{ii} - \Sigma_{ij}\Sigma_{jj}^{-1}\Sigma_{ji}, \Psi). \tag{3.19}$$

■

We next discuss some special cases. When $n_1 = 1$, $\tilde{\boldsymbol{Y}}_1 = \mathbf{Y}_0$ and $\tilde{\boldsymbol{\mu}}_1 = \boldsymbol{\mu}_0$. Σ_{11} becomes a positive number, denoted by σ_r^2, where subscript r denotes the row split case. Similar to the case in vector normal (equation (2.14)), write Σ_{22} as $\mathbf{R}_r$ and write Σ as

$$\Sigma = \sigma_r^2 \begin{pmatrix} 1 & \mathbf{r}_{0,r}^T \\ \mathbf{r}_{0,r} & \mathbf{R}_r \end{pmatrix}. \tag{3.20}$$

From the proof of the conditional distribution for partitioning by row (equation (3.18)), we can see for $n_1 = 1$ case, $\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{ji}$ will degenerate into a number $\sigma_r^2(1 - \mathbf{r}_{0,r}^T\mathbf{R}_r^{-1}\mathbf{r}_{0,r})$, and equation (3.18) (the conditional distribution) will turns into the usual vector normal,

$$\mathbf{Y}_0^T|\tilde{\boldsymbol{Y}}_2^T \sim \mathrm{N}_{s+1}(\boldsymbol{\mu}_0^T + (\tilde{\boldsymbol{Y}}_2^T - \tilde{\boldsymbol{\mu}}_2^T)\mathbf{R}_r^{-1}\mathbf{r}_{0,r}, \sigma_r^2(1 - \mathbf{r}_{0,r}^T\mathbf{R}_r^{-1}\mathbf{r}_{0,r})\Psi). \tag{3.21}$$

To partition by column, when $s_1 = 1$, $\tilde{\boldsymbol{Y}}_1 = \mathbf{Y}_0$ and $\tilde{\boldsymbol{\mu}}_1 = \boldsymbol{\mu}_0$, Ψ_{11} becomes a positive number, denoted by σ_c^2, where subscript c denotes the column partition case. Similar to above, write Ψ_{22} as $\mathbf{R}_c$ and write Ψ as

$$\Psi = \sigma_c^2 \begin{pmatrix} 1 & \mathbf{r}_{0,c}^T \\ \mathbf{r}_{0,c} & \mathbf{R}_c \end{pmatrix}. \tag{3.22}$$

From the conclusion of the conditional distribution for the column partition case (equation (3.9)), we can see for $s_1 = 1$ case, $\Psi_{11} - \Psi_{12}\Psi_{22}^{-1}\Psi_{21}$ becomes to one positive number $\sigma_c^2(1 - \mathbf{r}_{0,c}^T\mathbf{R}_c^{-1}\mathbf{r}_{0,c})$, and equation (3.9) (the conditional distribution) will turns into the usual vector normal

$$\mathbf{Y}_0|\tilde{\boldsymbol{Y}}_2 \sim \mathrm{N}_{n+1}(\boldsymbol{\mu}_0 + (\tilde{\boldsymbol{Y}}_2 - \tilde{\boldsymbol{\mu}}_2)\mathbf{R}_c^{-1}\mathbf{r}_{0,c}, \sigma_c^2(1 - \mathbf{r}_{0,c}^T\mathbf{R}_c^{-1}\mathbf{r}_{0,c})\Sigma). \tag{3.23}$$

3.2 Multivariate Gaussian Kriging Models

In this section, we first propose multivariate Gaussian kriging model. We will then derive the best linear unbiased prediction (BLUP) from the multivariate Gaussian kriging model.

3.2.1 Motivation

Let $y_j(\mathbf{x})$ be the j-th response at $\mathbf{x}$ for $j = 1, \cdots, s$. The ordinary Gaussian kriging model for $y_j(\mathbf{x})$ is defined to be

$$y_j(\mathbf{x}) = \mu_j + z_j(\mathbf{x}),$$

where μ_j is the unknown mean, and $z_j(\mathbf{x})$ is a random error. The Gaussian kriging model assumes that $z_j(\mathbf{x})$ is a Gaussian process with zero mean and covariance function

$$\mathrm{Cov}(z_j(\mathbf{s}), z_j(\mathbf{t})) = \sigma^2 r(\boldsymbol{\theta}; \mathbf{s}, \mathbf{t})$$

for some correlation function $r(\boldsymbol{\theta}; \mathbf{s}, \mathbf{t})$. For example,

$$r(\boldsymbol{\theta}; \mathbf{s}, \mathbf{t}) = \exp\{-\sum_{k=1}^{s} \theta_k (s_k - t_k)^2\}$$

and $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_s)^T$.

Now we generalize the univariate Gaussian kriging model for multiple response. Let $\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), \cdots, y_s(\mathbf{x}))$, $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_s)$ and $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), \cdots, z_s(\mathbf{x}))$. Thus,

$$\mathbf{y}(\mathbf{x}) = \boldsymbol{\mu} + \mathbf{z}(\mathbf{x}). \tag{3.24}$$

In this paper, we assume that $\mathbf{z}(\mathbf{x})$ is an s-dimensional Gaussian process indexed by $\mathbf{x}$. Specifically, for a given $\mathbf{x}$ in the experiment domain, $\mathbf{z}(\mathbf{x})$ follows $N_s(\mathbf{0}, \Psi)$, and for any $\mathbf{s}$ and $\mathbf{t}$, it is assumed that

$$\mathrm{Cov}(\mathbf{z}(\mathbf{s}), \mathbf{z}(\mathbf{t})) = r(\boldsymbol{\theta}; \mathbf{s}, \mathbf{t})\Psi. \tag{3.25}$$

for some correlation function $r(\boldsymbol{\theta}; \mathbf{s}, \mathbf{t})$. We call model (3.24) with the covariance function (3.25) to be multivariate Gaussian kriging model.

3.2.2 Prediction

Suppose that we collect data at $\mathbf{x}_1, \cdots, \mathbf{x}_n$. Let $\mathbf{Y}$ and $\mathbf{Z}$ be $n \times s$ matrices whose i-th rows are $\mathbf{y}(\mathbf{x}_i)^T$ and $\mathbf{z}(\mathbf{x}_i)^T$, respectively. Then,

$$\mathbf{Y} = \mathbf{1}_n \boldsymbol{\mu}^T + \mathbf{Z}.$$

More generally, we may replace $\mathbf{1}_n\boldsymbol{\mu}^T$ by $\mathbf{X}B$ to model the overall trend, where B is a $p \times s$ constant matrix and $\mathbf{X}$, an $n \times p$ matrix, consists of some basis functions that are used to capture overall trend. Thus,

$$\mathbf{Y} = \mathbf{X}B + \mathbf{Z}. \tag{3.26}$$

By the assumption on the covariance function of multivariate Gaussian kriging model, it follows that

$$\mathbf{Z} \sim \mathrm{N}_{n,s}(\mathbf{0}, \mathbf{R}, \Psi), \tag{3.27}$$

where $\mathrm{N}_{n,s}(\mathbf{0}, R, \Psi)$ stands for a matrix normal distribution, and $\mathbf{R}$ is an $s \times s$ matrix with (i, j)-element being $r(\boldsymbol{\theta}; \mathbf{x}_i, \mathbf{x}_j)$. The model (3.26) can be rewritten as

$$\mathbf{Y} \sim \mathrm{N}_{n,s}(\mathbf{X}B, \mathbf{R}, \Psi). \tag{3.28}$$

Using the matrix normal distribution theory (Muirhead, 1982), the conditional distribution of $\mathbf{y}_0 = \mathbf{y}(x_0)$ at untried point $\mathbf{x}_0$ given the information from $\mathbf{Y}$ is multivariate normal distribution with conditional mean

$$E(\mathbf{y}_0|\mathbf{Y}) = \mathbf{x}^*B + \mathbf{r}_0^T\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{X}B), \tag{3.29}$$

where $\mathbf{x}^*$ is a p-dimensional row vector corresponding the X-basis evaluated at $\mathbf{x}_0$ and $\mathbf{r}_0 = (r(\boldsymbol{\theta}; \mathbf{x}_1, \mathbf{x}_0), \cdots, r(\boldsymbol{\theta}, \mathbf{x}_n, \mathbf{x}_0))^T$, and conditional variance as

$$\mathrm{Var}(\mathbf{y}_0|\mathbf{Y}) = (1 - \mathbf{r}_0^T\mathbf{R}^{-1}\mathbf{r}_0)\Psi. \tag{3.30}$$

In the multivariate Gaussian kriging model, we predict $\mathbf{y}(\mathbf{x}_0)$ using the conditional mean $E(\mathbf{y}_0|\mathbf{Y})$. From the multivariate normal distribution theory, we know that the prediction from the multivariate Gaussian kriging model has the following properties:

1. The prediction (3.29) is the best linear unbiased predictor (BLUP) under the assumption of normality;

2. Gaussian kriging prediction (3.29) interpolates the observed data.

Since B, Ψ and $\boldsymbol{\theta}$ are unknown, we substitute the unknown parameters with their maximum likelihood estimate derived in Section 3.3. Thus, the prediction (3.29) will be

$$\hat{\mathbf{y}}_0 = \mathbf{x}^*\hat{B} + \mathbf{r}_0^T(\hat{\boldsymbol{\theta}})\mathbf{R}^{-1}(\hat{\boldsymbol{\theta}})(\mathbf{Y} - \mathbf{X}\hat{B}), \tag{3.31}$$

and the corresponding variance-covariance matrix for the prediction is

$$\mathrm{Var}(\hat{\mathbf{y}}_0|\mathbf{Y}) = \{1 - \mathbf{r}_0^T(\hat{\boldsymbol{\theta}})\mathbf{R}^{-1}(\hat{\boldsymbol{\theta}})\mathbf{r}_0(\hat{\boldsymbol{\theta}})\}\hat{\Psi}. \tag{3.32}$$

3.3 Estimation of parameters in the matrix case

In this section, we will present the parameter estimation procedure for multivariate Gaussian kriging model. We will first present the MLE (and penalized MLE), and followed by cross-validation method.

3.3.1 Maximum Likelihood Estimator

In this part, we extend maximum likelihood estimation for Gaussian kriging models to the case of multiple response. Since in computer experiment, we usually don't have enough observation to estimate all the parameters separately, we assume **R** and Ψ have the following relationship:

$$\Psi = \frac{1}{n}\mathbf{E}^T\mathbf{R}^{-1}\mathbf{E}, \tag{3.33}$$

where $\mathbf{E} = \mathbf{Y} - \mathbf{X}B$.

We first derive the likelihood function for the observations **Y**. Rewrite the model (3.28) in vector form

$$vec(\mathbf{Y}) \sim \mathrm{N}_{ns}(vec(\mathbf{X}B), \mathbf{R} \otimes \Psi). \tag{3.34}$$

The likelihood function for $vec(\mathbf{Y})$ is

$$\begin{aligned}
& \mathrm{L}(B, \mathbf{R}, \Psi) \\
= \ & (2\pi)^{\frac{ns}{2}} |\mathbf{R} \otimes \Psi|^{-\frac{1}{2}} \exp(-\frac{1}{2}(vec\mathbf{Y} - vec(\mathbf{X}B))^T (\mathbf{R} \otimes \Psi)^{-1} (vec\mathbf{Y} - vec(\mathbf{X}B))) \\
= \ & (2\pi)^{\frac{ns}{2}} |\mathbf{R}|^{-\frac{s}{2}} |\Psi|^{-\frac{n}{2}} \exp(-\frac{1}{2}[vec(\mathbf{Y} - \mathbf{X}B)]^T [\mathbf{R}^{-1} \otimes \Psi^{-1}][vec(\mathbf{Y} - \mathbf{X}B)]) \\
= \ & (2\pi)^{\frac{ns}{2}} |\mathbf{R}|^{-\frac{s}{2}} |\Psi|^{-\frac{n}{2}} \exp(-\frac{1}{2} tr((\mathbf{Y} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}B)\Psi^{-1})). \qquad (3.35)
\end{aligned}$$

After dropping some constants, the log-likelihood becomes

$$l(B, \mathbf{R}, \Psi) = -\tfrac{s}{2} \log |\mathbf{R}| - \tfrac{n}{2} log|\Psi| - \tfrac{1}{2} tr((\mathbf{Y} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}B)\Psi^{-1}) \qquad (3.36)$$

The MLE to maximize the above likelihood function for the coefficient matrix B is

$$\hat{B} = (\mathbf{X}^T \mathbf{R}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{R}^{-1} \mathbf{Y}, \qquad (3.37)$$

which can be derived from the following arguments

$$\begin{aligned}
& (\mathbf{Y} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}B) \Psi^{-1} \\
= \ & (\mathbf{Y} - \mathbf{X}\hat{B} + \mathbf{X}\hat{B} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}\hat{B} + \mathbf{X}\hat{B} - \mathbf{X}B) \Psi^{-1} \\
= \ & (\mathbf{Y} - \mathbf{X}\hat{B})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}\hat{B}) \Psi^{-1} + (\mathbf{X}\hat{B} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{X}\hat{B} - \mathbf{X}B) \Psi^{-1} \\
& + (\mathbf{Y} - \mathbf{X}\hat{B})^T \mathbf{R}^{-1} (\mathbf{X}\hat{B} - \mathbf{X}B) \Psi^{-1} + (\mathbf{X}\hat{B} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}\hat{B}) \Psi^{-1}.
\end{aligned}$$

After plugging $\hat{B}$ in (3.37) into the last two terms in the above expression, we can easily found that these two terms are zero. Then the last term in equation (3.36) turns into

$$\begin{aligned}
& \tfrac{1}{2} \tfrac{tr((\mathbf{Y} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}B)\Psi^{-1})}{2} \\
= \ & \tfrac{1}{2} \tfrac{tr((\mathbf{Y} - \mathbf{X}\hat{B})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}\hat{B})\Psi^{-1})}{2} + \tfrac{1}{2} \tfrac{tr((\mathbf{X}\hat{B} - \mathbf{X}B)^T \mathbf{R}^{-1} (\mathbf{X}\hat{B} - \mathbf{X}B)\Psi^{-1})}{2} \qquad (3.38) \\
= \ & \tfrac{1}{2} \tfrac{tr((\mathbf{Y} - \mathbf{X}\hat{B})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{X}\hat{B})\Psi^{-1})}{2} + \tfrac{1}{2} \tfrac{[vec(\mathbf{X}\hat{B} - \mathbf{X}B)]^T [\mathbf{R} \otimes \Psi]^{-1} [vec(\mathbf{X}\hat{B} - \mathbf{X}B)]}{2}.
\end{aligned}$$

Since the variance-covariance matrix $\mathbf{R}$ and Ψ are symmetric and we assume $\mathbf{R} > 0$ and $\Psi > 0$, all the eigenvalues for these two matrix are positive. But the eigenvalues for $\mathbf{R} \otimes \Psi$ is the product of the eigenvalues of these two matrix, hence $\mathbf{R} \otimes \Psi > 0$ hence $[\mathbf{R} \otimes \Psi]^{-1} > 0$. Then the last term in the above equation is always greater than or equal to zero. It is equal to zero for any matrix $\mathbf{X}$ if and only if $B = \hat{B}$. So the likelihood function achieves its maximum at $B = \hat{B}$. This proves the MLE for the coefficient matrix B is $\hat{B}$ in (3.37).

The MLE for $\boldsymbol{\theta}$ does not have a closed form. We will need to use Newton-Raphson algorithm to estimate it. At the MLE $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, we have

$$\frac{\partial l}{\partial \boldsymbol{\theta}}|_{\hat{\boldsymbol{\theta}}} = \mathbf{0}. \tag{3.39}$$

For any estimation $\boldsymbol{\theta}^{(0)}$ which is close to the MLE $\hat{\boldsymbol{\theta}}$, we can use Taylor's expansion

$$\frac{\partial l}{\partial \boldsymbol{\theta}}|_{\hat{\boldsymbol{\theta}}} = \frac{\partial l}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}^{(0)}} + \frac{\partial^2 l}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{(0)}) + o_p(\|(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{(0)}\}). \tag{3.40}$$

Hence the updated estimation is

$$\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}^{(0)} - (\frac{\partial^2 l}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T})^{-1} \frac{\partial l}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}^{(0)}}. \tag{3.41}$$

Similar to the estimating procedure in the vector case, the following estimating procedures can be employed to get the estimated coefficient matrix and all the parameters in the variance-covariance matrix:

Step 1 Set the initial value of B to be $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$, the ordinary least square estimates of B in model (3.26);

Step 2 For a given B, update $\boldsymbol{\theta}$ using (3.39);

Step 3 For given $\boldsymbol{\theta}$, update B using (3.37);

Step 4 Iterate step 2 and step 3 until it converges.

3.3.2 First and second derivatives of the likelihood function with respect to the parameters in covariance matrix

In order to use the Fisher scoring algorithm to get the MLE for B and $\boldsymbol{\theta}$, we need to calculate the first and second derivatives of the likelihood function. The following matrix calculation will be utilized to get these derivatives.

For a matrix $\mathbf{A}$ which is a matrix function of vector $\boldsymbol{\theta}$, we are interested in calculating $\frac{\partial \mathbf{A}^{-1}(\boldsymbol{\theta})}{\partial \theta_j}$ and $\frac{\partial log|\mathbf{A}(\boldsymbol{\theta})|}{\partial \theta_j}$. We know that

$$\mathbf{A}^{-1}(\boldsymbol{\theta})\mathbf{A}(\boldsymbol{\theta}) = \mathbf{I}.$$

Taking the derivative for both sides with respect to each component of $\boldsymbol{\theta}$, we have

$$\frac{\partial \mathbf{A}^{-1}(\boldsymbol{\theta})}{\partial \theta_j}\mathbf{A}(\boldsymbol{\theta}) + \mathbf{A}^{-1}(\boldsymbol{\theta})\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_j} = \mathbf{0}.$$

Thus,

$$\frac{\partial \mathbf{A}^{-1}(\boldsymbol{\theta})}{\partial \theta_j} = -\mathbf{A}^{-1}(\boldsymbol{\theta})\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_j}\mathbf{A}^{-1}(\boldsymbol{\theta}). \tag{3.42}$$

Next, we derive $\frac{\partial log|\mathbf{A}(\boldsymbol{\theta})|}{\partial \theta_j}$:

$$\begin{aligned}\frac{\partial log|\mathbf{A}(\boldsymbol{\theta})|}{\partial \theta_j} &= \frac{1}{|\mathbf{A}(\boldsymbol{\theta})|}tr(\frac{\partial |\mathbf{A}(\boldsymbol{\theta})|}{\partial \mathbf{A}(\boldsymbol{\theta})}\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_j}) \qquad (3.43)\\ &= \frac{1}{|\mathbf{A}(\boldsymbol{\theta})|}tr(|\mathbf{A}(\boldsymbol{\theta})|\mathbf{A}^{-1}(\boldsymbol{\theta})\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_j}) = tr(\mathbf{A}^{-1}(\boldsymbol{\theta})\frac{\partial \mathbf{A}(\boldsymbol{\theta})}{\partial \theta_j}).\end{aligned}$$

To calculate the derivatives of the log-likelihood function, we rewrite (3.36) at $B = \hat{B}$

$$l(B, \boldsymbol{\theta})|_{B=\hat{B}} = -\frac{s}{2}log|\mathbf{R}(\boldsymbol{\theta})| - \frac{n}{2}log|\Psi| - \frac{tr((\mathbf{Y}-\mathbf{X}\hat{B})^T\mathbf{R}^{-1}(\mathbf{Y}-\mathbf{X}\hat{B})\Psi^{-1})}{2} \tag{3.44}$$

Denote

$$\mathbf{E} = \mathbf{Y} - \mathbf{X}\hat{B}$$

as the residual matrix for a given estimator of the coefficient matrix $\hat{B}$, and substitute (3.33) into the above formula, then the log-likelihood function will become

$$l(B, \boldsymbol{\theta})|_{B=\hat{B}} = -\frac{s}{2}log|\mathbf{R}| - \frac{n}{2}log|\Psi| - \frac{s}{2}. \tag{3.45}$$

Denote j and k as the component index for vector $\boldsymbol{\theta}$, $\dot{l}_j = \frac{\partial l}{\partial \theta_j}$, $\ddot{l}_{jk} = \frac{\partial^2 l}{\partial \theta_j \partial \theta_k}$, $\dot{\mathbf{R}}_j = \frac{\partial \mathbf{R}}{\partial \theta_j}$, $\ddot{\mathbf{R}}_{jk} = \frac{\partial^2 \mathbf{R}}{\partial \theta_j \partial \theta_k}$, employ equations (3.42) and (3.43), it follows that

$$\begin{aligned}
\dot{l}_j &= -\tfrac{s}{2}tr(\mathbf{R}^{-1}\dot{\mathbf{R}}_j) - \tfrac{n}{2}tr(\Psi^{-1}\dot{\Psi}_j) \\
&= \tfrac{1}{2}tr(\Psi^{-1}\mathbf{E}^T\mathbf{R}^{-1}\dot{\mathbf{R}}_j\mathbf{R}^{-1}\mathbf{E}) - \tfrac{s}{2}tr(\mathbf{R}^{-1}\dot{\mathbf{R}}_j) \\
&= \tfrac{1}{2}tr(\Psi^{-1}\mathbf{E}^T\mathbf{R}^{-1}\dot{\mathbf{R}}_j\mathbf{R}^{-1}\mathbf{E} - s\mathbf{R}^{-1}\dot{\mathbf{R}}_j) \\
&= \tfrac{1}{2}tr(\mathbf{R}^{-1}(\mathbf{E}\Psi^{-1}\mathbf{E}^T - s\mathbf{R})\mathbf{R}^{-1}\dot{\mathbf{R}}_j).
\end{aligned}$$

Denote $\phi = \mathbf{E}\Psi^{-1}\mathbf{E}^T$, $\eta = \mathbf{R}^{-1}\phi\mathbf{R}^{-1}$, then

$$\dot{l}_j = \frac{1}{2}tr((\eta - s\mathbf{R}^{-1})\dot{\mathbf{R}}_j).$$

The second derivative of the log-likelihood function is

$$\begin{aligned}
\ddot{l}_{jk} &= -\tfrac{1}{2}tr(\mathbf{R}^{-1}\dot{\mathbf{R}}_k\eta\dot{\mathbf{R}}_j) - \tfrac{1}{2}tr(\mathbf{R}^{-1}\mathbf{E}\Psi^{-1}\dot{\Psi}\Psi^{-1}\mathbf{E}^T\mathbf{R}^{-1}\dot{\mathbf{R}}_j) \\
&\quad -\tfrac{1}{2}tr(\eta\dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j) + \tfrac{1}{2}tr(\eta\ddot{\mathbf{R}}_{jk}) + \tfrac{s}{2}tr(\mathbf{R}^{-1}\dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j) - \tfrac{s}{2}tr(\mathbf{R}^{-1}\ddot{\mathbf{R}}_{jk}) \\
&= \tfrac{1}{2}tr(\eta(\ddot{\mathbf{R}}_{jk} + \tfrac{\dot{\mathbf{R}}_k\eta\dot{\mathbf{R}}_j}{n} - \dot{\mathbf{R}}_j\mathbf{R}^{-1}\dot{\mathbf{R}}_k - \dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j)) \\
&\quad +\tfrac{s}{2}tr(\mathbf{R}^{-1}(\dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j - \ddot{\mathbf{R}}_{jk})).
\end{aligned}$$

In summary, we have

$$\dot{l}_j = \frac{1}{2}tr((\eta - s\mathbf{R}^{-1})\dot{\mathbf{R}}_j), \tag{3.46}$$

and

$$\begin{aligned}
\ddot{l}_{jk} &= \tfrac{1}{2}tr(\eta(\ddot{\mathbf{R}}_{jk} + \tfrac{\dot{\mathbf{R}}_k\eta\dot{\mathbf{R}}_j}{n} - \dot{\mathbf{R}}_j\mathbf{R}^{-1}\dot{\mathbf{R}}_k - \dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j)) \\
&\quad +\tfrac{s}{2}tr(\mathbf{R}^{-1}(\dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j - \ddot{\mathbf{R}}_{jk})).
\end{aligned} \tag{3.47}$$

Apparently, all the cross-second-derivative have expectation zero for B and $\boldsymbol{\theta}$. Hence we can use iteration algorithm to get the MLE for all these parameters. Let $\dot{\mathbf{l}}_r = (\dot{l}_j)$, $\mathbf{I} = -(\ddot{l}_{jk})$, then the iteration algorithm is given below

Step 1 Set initial value of B to be $\hat{B}^{(0)} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$, the ordinary least square estimates of B in model (3.26), set some initial values for $\hat{\boldsymbol{\theta}}^{(0)}$;

Step 2 Calculate $\mathbf{E}^{(0)} = \mathbf{Y} - \mathbf{X}\hat{B}^{(0)}$.

Step 3 Update $\hat{\boldsymbol{\theta}}^{(1)} = \hat{\boldsymbol{\theta}}^{(0)} + \mathbf{I}^{-1}\dot{\mathbf{l}}$

Step 4 Update $\mathbf{R} = \mathbf{R}(\hat{\boldsymbol{\theta}}^{(1)})$

Step 5 Update $\hat{B}^{(1)} = (\mathbf{X}^T\mathbf{R}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}^{-1}\mathbf{Y}$

Step 6 Check convergence. If not, go to step 2 and repeat.

3.3.3 MLE by Penalized Likelihood with L_2 Penalty

When the likelihood function does not has any mode, the MLE is difficult to get. For the case that the likelihood function will increase with the increase of the parameters, the MLE from penalized likelihood is an option. Here, we will use an L_2 penalty because of the simplicity and the straightforward usage. See Li and Sudjianto (2005) for detailed illustrations on penalized Guassian kriging model with other penalties.

To penalize the effect of the increasing parameters, the L_2 norm of the parameters will be subtracted from the original log-likelihood equation (3.45)

$$l(B, \boldsymbol{\theta})|_{B=\hat{B}} = -\frac{s}{2}log|\mathbf{R}| - \frac{n}{2}log|\Psi| - \frac{s}{2} - \frac{\|\boldsymbol{\theta}\|^2}{\sqrt{n\log n}}. \tag{3.48}$$

The first and second derivative of the penalized loglikelihood function turn into:

$$\dot{l}_j = \frac{1}{2}tr((\eta - s\mathbf{R}^{-1})\dot{\mathbf{R}}_j) - \frac{2\theta_j}{\sqrt{n\log n}}, \tag{3.49}$$

and

$$\begin{aligned} \ddot{l}_{jk} &= \tfrac{1}{2}tr(\eta(\ddot{\mathbf{R}}_{jk} + \tfrac{\dot{\mathbf{R}}_k\eta\dot{\mathbf{R}}_j}{n} - \dot{\mathbf{R}}_j\mathbf{R}^{-1}\dot{\mathbf{R}}_k - \dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j)) \\ &\quad + \tfrac{s}{2}tr(\mathbf{R}^{-1}(\dot{\mathbf{R}}_k\mathbf{R}^{-1}\dot{\mathbf{R}}_j - \ddot{\mathbf{R}}_{jk})) - \tfrac{2\delta_{jk}}{\sqrt{n\log n}}, \end{aligned} \tag{3.50}$$

where $\delta_{jk} = \begin{cases} 1 & if \quad j = k \\ 0 & o.w. \end{cases}$.

3.3.4 Estimator by Cross-Validation

Although MLE has many good properties over other estimators, the implement in the real case data may have some severe problems. First of all, due to the data itself, the Hessian matrix required in the Fisher scoring algorithm will sometimes be singular or close to singular. The condition number for this matrix will become fairly large hence the computation result will be unstable. Secondly, if the likelihood function is not unimodal, it is very hard to find MLE. It is especially difficult for the likelihood function that will flatten out on the tail and does not have a maximum point. For the single parameter case, it will be easier to use other data-driven methods such as cross-validation to estimate the parameters.

Consider the single parameter case for variable θ. If the number of observations is small, then we will use leave-one-out cross validation. Suppose we have n observations, for a given value of θ, we sequentially hide one of them (leave one out), and use the rest $n-1$ observations to construct the kriging model. Then the kriging model will pass through every point of these $n-1$ observations and will not agree at the one we hide. Denote the prediction at the ith point for leaving the ith observation point out from the Kriging model as $\mathbf{Y}_i^{-i}$, then we accumulate this disagreement by adding the square of the residual $\mathbf{Y}_i - \mathbf{Y}_i^{-i}$ as the CV (cross validation) score

$$CV = \sum_{i=1}^{n} (\mathbf{Y}_i - \mathbf{Y}_i^{-i})(\mathbf{Y}_i - \mathbf{Y}_i^{-i})^T. \tag{3.51}$$

This CV score actually measures the prediction ability under the current parameter value. We will pick the parameter value which corresponding to the smallest CV score.

3.4 Functional ANOVA

From (3.31), we obtain a metamodel

$$\mathbf{g}(\mathbf{x}_0)\hat{=}\hat{\mathbf{y}}(\mathbf{x}_0) = \mathbf{x}^*\hat{B} + \mathbf{r}_0^T(\hat{B}))\mathbf{R}^{-1}(\hat{\boldsymbol{\theta}})(\mathbf{Y} - \mathbf{X}B), \tag{3.52}$$

where $\mathbf{g}(\mathbf{x}) = (g^{(1)}(\mathbf{x}), \cdots, g^{(s)}(\mathbf{x}))$ is a s-dimensional function. Let $\mathbf{M} = \hat{\mathbf{R}}^{-1}(\mathbf{Y} - \mathbf{X}\hat{B}) = (\mathbf{M}_1^T, \cdots, \mathbf{M}_n^T)^T$ where $\mathbf{M}$ is an n by s matrix and $\mathbf{M}_i$s are 1 by s vectors. Then the final kriging model becomes

$$\hat{\mathbf{Y}}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) = \mathbf{x}^*\hat{B} + \sum_{i=1}^{n} r_i(\mathbf{x})\mathbf{M}_i, \tag{3.53}$$

where

$$r_i(\mathbf{x}) = exp(-\sum_{j=1}^{d} \hat{\theta}_j(x_j - X_{i,j})^2). \tag{3.54}$$

As discussed in Chapter 2, we know that the kriging model (3.52) smoothly passes through each observation point. Now we want to know how and how much each variable affects the response. This is very similar to the decomposition of sum of squares in ANOVA. Here, we use Functional ANOVA to decompose this function (3.52) onto each variable direction.

Suppose a single value function $g(\mathbf{x})$ is integrable, it can be shown (Sobol′ (1993), Sobol′ (2001) and Sobol′ (2003)) that this function can be decomposed into

$$g(\mathbf{x}) = g_0 + \sum_{i=1}^{d} g_i(x_i) + \sum_{i<j} g_{ij}(x_i, x_j) + \cdots + g_{1\cdots d}(x_1, \cdots, x_d), \tag{3.55}$$

where

$$\int g_{i_1\cdots i_t}(x_{i_1}, \cdots, x_{i_t})dx_k = 0, \quad k = i_1, \cdots, i_t, \tag{3.56}$$

and all the summands are orthogonal.

Since $\mathbf{g}$ is a integrable function in kriging model (3.52), it is straight forward to show that the multivariate function $\mathbf{g}$ can also be decomposed into the similar form:

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}_0 + \sum_{i=1}^{d} \mathbf{g}_i(x_i) + \sum_{i<j} \mathbf{g}_{ij}(x_i, x_j) + \cdots + \mathbf{g}_{1\cdots s}(x_1, \cdots, x_s), \tag{3.57}$$

where $\mathbf{g}_0 = (g_0^{(1)}, \cdots, g_0^{(s)})$, $\mathbf{g}_i(x_i) = (g_i^{(1)}(x_i), \cdots, g_i^{(s)}(x_i))$, etc. And

$$\int \mathbf{g}_{i_1 \cdots i_t}(x_{i_1}, \cdots, x_{i_t}) dx_k = \mathbf{0}, \quad k = i_1, \cdots, i_t, \tag{3.58}$$

and all the summands are orthogonal in the sense that

$$\int g^{(l)}_{i_1 \cdots i_u} g^{(l)}_{j_1 \cdots j_v} d\mathbf{x} = 0, \quad l = 1, \cdots, s, \tag{3.59}$$

where $(i_1, \cdots, i_u) \neq (j_1, \cdots, j_v)$.

Integrating (3.57) and using the above properties, we have

$$E\mathbf{g} = \int \mathbf{g}(\mathbf{x}) d\mathbf{x} = \mathbf{g}_0,$$

$$E\mathbf{g}(\mathbf{X}|X_i = x_i) = \int \mathbf{g}(\mathbf{x}) \Pi_{k \neq i} dx_k = \mathbf{g}_0 + \mathbf{g}_i(x_i),$$

and so on, where $\mathbf{g}_0$ is a constant 1 by s vector and $\mathbf{g}_i$s are 1 by s functional vectors. In this functional ANOVA decomposition, $\mathbf{g}_0$ can be viewed as the overall effect, and $\mathbf{g}_i$s can be viewed as the main effect for each variable x_i. In this case, we only analyze the data up to the main effect. The result shows that this main effect account for 96% of the total variance.

It can be easily seen that $\mathbf{g}(\mathbf{x})$ is also square integrable. Due to the orthogonality of the decomposition in (3.57), for each component of $\mathbf{g}$, we have

$$\begin{aligned} E[g^{(l)}]^2 = \int {g^{(l)}}^2 d\mathbf{x} &= {g_0^{(l)}}^2 + \textstyle\sum_{i=1}^d \int {g_i^{(l)}}^2(x_i) dx_i + \sum_{i<j} \int {g_{ij}^{(l)}}^2(x_i, x_j) dx_i dx_j \\ & \quad + \cdots + \textstyle\int {g_{1 \cdots d}^{(l)}}^2 (x_1, \cdots, x_d) dx_i \cdots dx_d. \end{aligned}$$

for $l = 1, \cdots, s$.

Define the componentwise total variance and the total variance by

$$D^{(l)} = \text{Var} g^{(l)} = \int {g^{(l)}}^2(\mathbf{x}) d\mathbf{x} - {g_0^{(l)}}^2 \tag{3.60}$$

and

$$D = \sum_{l=1}^{s} \text{Var} g^{(l)} = \sum_{l=1}^{s} D^{(l)}. \tag{3.61}$$

Also define the componentwise partial variances and partial variances by

$$D^{(l)}_{i_1\cdots i_t} = \text{Var} g^{(l)}_{i_1\cdots i_t} = \int g^{(l)2}_{i_1\cdots i_t}(x_{i_1}, \cdots, x_{i_t})dx_{i_1}\cdots dx_{i_t}, \tag{3.62}$$

and

$$D_{i_1\cdots i_t} = \sum_{l=1}^{s} D^{(l)}_{i_1\cdots i_t}. \tag{3.63}$$

Note that

$$D^{(l)} = \sum_{k=1}^{d} \sum_{i_1<\cdots<i_k} D^{(l)}_{i_1\cdots i_k}$$

and

$$D = \sum_{k=1}^{d} \sum_{i_1<\cdots<i_k} D_{i_1\cdots i_k}$$

respectively, which is similar to the decomposition in traditional ANOVA.

We extend the Sobol′ indices to componentwise Sobol′ indices and overall Sobol′ indices as

$$S^{(l)}_{i_1\cdots i_k} = \frac{D^{(l)}_{i_1\cdots i_k}}{D^{(l)}},$$

and

$$S_{i_1\cdots i_k} = \frac{D_{i_1\cdots i_k}}{D}.$$

These indices reflect how much does each variable affect the response.

In order to calculate the various Sobol′ indices, we need to integrate the kriging model (3.52). We illustrate this integration as follows. Let $\Phi(\cdot)$ be the cumulative distribution function of $N(0,1)$. For normalized predictor variables $x_1, \cdots, x_{18} \in (0,1)^d$, we have

$$\begin{aligned}
\int_{\mathbf{X}\in[0,1]^d} r_i d\mathbf{X} &= \int_{\mathbf{X}\in[0,1]^d} exp(-\textstyle\sum_{k=1}^d \hat{\theta}_k(x_k - X_{i,k})^2)dx_1\cdots dx_d \\
&= \int_{\mathbf{X}\in[0,1]^d} \Pi_{k=1}^d exp(-\hat{\theta}_k(x_k - X_{i,k})^2)dx_1\cdots dx_d \\
&= \Pi_{k=1}^d \int_0^1 exp(-\hat{\theta}_k(x_k - X_{i,k})^2)dx_k \\
&= \Pi_{j=1}^d \tfrac{\pi}{\hat{\theta}_j} \int_{-X_{i,j}\sqrt{2\hat{\theta}_j}}^{(1-X_{i,j})\sqrt{2\hat{\theta}_j}} \tfrac{1}{2\pi} exp(-x^2)dx \\
&= \Pi_{j=1}^d \tfrac{\pi}{\hat{\theta}_j}(\Phi((1-X_{i,j})\sqrt{2\hat{\theta}_j}) - \Phi(-X_{i,j}\sqrt{2\hat{\theta}_j})),
\end{aligned} \tag{3.64}$$

$$\begin{aligned} r_i r_j &= exp(-\textstyle\sum_{k=1}^{d} \hat{\theta}_k (x_k - X_{i,k})^2) exp(-\textstyle\sum_{k=1}^{d} \hat{\theta}_k (x_k - X_{j,k})^2) \\ &= \Pi_{k=1}^{d} exp(-\hat{\theta}_k ((x_k - X_{i,k})^2 + (x_k - X_{j,k})^2) \\ &= \Pi_{k=1}^{d} exp(-2\hat{\theta}_k [(x_k - \tfrac{X_{i,k}+X_{j,k}}{2})^2 + (\tfrac{X_{i,k}-X_{j,k}}{2})^2) \\ &= exp(-\tfrac{1}{2} \textstyle\sum_{k=1}^{d} \hat{\theta}_k (X_{i,k} - X_{j,k})^2) \Pi_{k=1}^{d} exp(-\frac{(x_k - \frac{X_{i,k}+X_{j,k}}{2})^2}{2\frac{1}{4\hat{\theta}_k}}), \end{aligned} \tag{3.65}$$

$$\begin{aligned} \textstyle\int_{\mathbf{x}\in[0,1]^d} r_i r_j d\mathbf{x} &= exp(-\tfrac{1}{2} \textstyle\sum_{k=1}^{d} \hat{\theta}_k (X_{i,k} - X_{j,k})^2) \\ &\quad \Pi_{k=1}^{d} \textstyle\int_0^1 exp(-\frac{(x_k - \frac{X_{i,k}+X_{j,k}}{2})^2}{2\frac{1}{4\hat{\theta}_k}}) dx_k \\ &= exp(-\tfrac{1}{2} \textstyle\sum_{k=1}^{d} \hat{\theta}_k (X_{i,k} - X_{j,k})^2) \\ &\quad \Pi_{k=1}^{d} \sqrt{\tfrac{\pi}{2\hat{\theta}_k}} \textstyle\int_{-(X_{i,k}+X_{j,k})\sqrt{\hat{\theta}_k}}^{2(1-\frac{X_{i,k}+X_{j,k}}{2})\sqrt{\hat{\theta}_k}} \frac{1}{\sqrt{2\pi}} exp(-\tfrac{x^2}{2}) dx \\ &= \Pi_{k=1}^{d} \tfrac{\pi}{2\hat{\theta}_k} [\Phi(2(1 - \tfrac{X_{i,k}+X_{j,k}}{2})\sqrt{\hat{\theta}_k}) - \Phi(-(X_{i,k} + X_{j,k})\sqrt{\hat{\theta}_k})] \\ &\quad exp(-\tfrac{1}{2} \textstyle\sum_{k=1}^{d} \hat{\theta}_k (X_{i,k} - X_{j,k})^2), \end{aligned} \tag{3.66}$$

$$\begin{aligned} \mathbf{g}_0 = E\mathbf{g}_0 &= \textstyle\int_{\mathbf{x}\in[0,1]^d} \mathbf{g} d\mathbf{x} \\ &= \hat{\boldsymbol{\beta}}_0 + \textstyle\int_{\mathbf{x}\in[0,1]^d} \sum_{i=1}^{d} x_i \hat{\boldsymbol{\beta}}_i d\mathbf{x} + \sum_{i=1}^{n} \int_{\mathbf{x}\in[0,1]^d} r_i \mathbf{M}_i d\mathbf{x} \\ &= \hat{\boldsymbol{\beta}}_0 + \tfrac{1}{2} \textstyle\sum_{i=1}^{d} \hat{\boldsymbol{\beta}}_i + \sum_{i=1}^{n} \int_{\mathbf{x}\in[0,1]^d} r_i \mathbf{M}_i d\mathbf{x}, \end{aligned} \tag{3.67}$$

$$\begin{aligned} \mathbf{g}^2 &= \hat{\boldsymbol{\beta}}_0^2 + 2\hat{\boldsymbol{\beta}}_0 \textstyle\sum_{i=1}^{d} x_i \hat{\boldsymbol{\beta}}_i + \sum_{i=1}^{d} \sum_{j=1}^{d} x_i x_j \hat{\boldsymbol{\beta}}_i \hat{\boldsymbol{\beta}}_j \\ &\quad +2(\hat{\boldsymbol{\beta}}_0 + \textstyle\sum_{j=1}^{d} x_j \hat{\boldsymbol{\beta}}_j) \sum_{i=1}^{n} \mathbf{M}_i r_i + \sum_{i=1}^{n} \sum_{j=1}^{n} r_i r_j \mathbf{M}_i \mathbf{M}_j, \end{aligned} \tag{3.68}$$

$$\begin{aligned} E\mathbf{g}^2 &= \textstyle\int_{\mathbf{x}\in[0,1]^d} \mathbf{g}^2 d\mathbf{x} \\ &= \hat{\boldsymbol{\beta}}_0^2 + \hat{\boldsymbol{\beta}}_0 \textstyle\sum_{i=1}^{d} \hat{\boldsymbol{\beta}}_i + \frac{1}{3} \sum_{i=1}^{d} \hat{\boldsymbol{\beta}}_i + \frac{1}{2} \sum_{i>j} \sum \hat{\boldsymbol{\beta}}_i \hat{\boldsymbol{\beta}}_j \\ &\quad +2\hat{\boldsymbol{\beta}}_0 \textstyle\sum_{i=1}^{d} \mathbf{M}_i \int_{\mathbf{x}\in[0,1]^d} r_i d\mathbf{x} + \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{M}_i \mathbf{M}_j \int_{\mathbf{x}\in[0,1]^d} r_i r_j d\mathbf{x} \\ &\quad +2 \textstyle\sum_{j=1}^{d} \hat{\boldsymbol{\beta}}_j \sum_{i=1}^{n} \mathbf{M}_i \int_{\mathbf{x}\in[0,1]^d} x_j r_i d\mathbf{x}. \end{aligned} \tag{3.69}$$

Except for the last integration term in (3.69), we can get the analytical result for all of these terms. We use Monte-Carlo simulation to get the last integration in

(3.69). Finally, the ith decomposed linear term is

$$\begin{aligned} \mathbf{g}_i(x_i) &= \int \mathbf{g} dx_1 \cdots dx_{i-1} dx_{i+1} \cdots dx_d - E\mathbf{g} \\ &= (x_i - \tfrac{1}{2})\boldsymbol{\beta}_i + \sum_{k=1}^{n} \mathbf{M}_i \int r_k dx_1 \cdots dx_{k-1} dx_{k+1} \cdots dx_d - \sum_{k=1}^{n} \mathbf{M}_i \int r_k d\mathbf{x} \end{aligned} \tag{3.70}$$

3.5 Simulation Study

To validate our code, we perform the following simulation. We take the design matrix **X** the same as the one which is corresponding to the 30 design cases in Example 1.1. The coefficient matrix $\boldsymbol{\beta}$ is listed in table 3.1. The error term needs to be a smooth surface, which is taken to be

$$\epsilon_{ij} = c[\sin(2k\pi t_j) + \cos(2k\pi t_j)]exp(-\frac{1}{2}\mathbf{x}_i \mathbf{A}_j \mathbf{x}_i'), \tag{3.71}$$

where $i = 1, 2, \cdots, 30$, $t_j \in (0.3, 0.7)$ for $j = 1, 2, 3, 4, 5$, and $\mathbf{A}_j$ is a square matrix with off diagonal elements equal to t_j, and diagonal element equal to unit. $\mathbf{A}_j$ can be viewed as the correlation matrix for $\mathbf{x}_i$ corresponding to the jth response. In this simulation study, we take $k = 12$. c controls the error-signal ratio. In order to see the effect of the error level to our code result, c is taken to be the following values: 0.01,0.1,0.5,1,2,5,10. The true model for this simulation is

$$y_{ij} = \mathbf{x}_i \boldsymbol{\beta}_j + \epsilon_{ij}. \tag{3.72}$$

After the response is generated according to model (3.72), we use Fisher scoring algorithm which is discussed in Section 3.3 & 3.4 to get the estimation for the parameters in Gaussian kriging model. In order to measure the code quality under different noise level, we define the mean-square-error (MSE) for individual vector coefficient $\boldsymbol{\beta}_i$ as: $MSE_i = \frac{1}{5}\sum_{j=1}^{5}(\beta_{ij} - \hat{\beta}_{ij})^2$. Table 3.2 gives this MSE result. We can see that with the increase of the error level c, the MSE is increasing. The MSE cannot give us a whole picture of the accuracy of the model. Table 3.3 gives the error percentage for individual coefficient elements for the largest noise level case $c = 10$. We

Table 3.1. Coefficient matrix for simulation of multivariate kriging model

	1	2	3	4	5
β_0	1.4548	0.9415	1.2267	0.9575	0.6039
β_1	-0.0444	0.0611	-0.0241	0.4092	-0.0361
β_2	0.0730	0.1179	0.2787	-0.0320	-0.3914
β_3	0.5460	0.0477	-0.4631	0.3292	0.7299
β_4	-0.0953	-0.0612	-0.3454	-0.5185	-0.0385
β_5	0.1371	0.1864	-0.7279	0.2930	0.0342
β_6	0.0180	-0.0905	-0.5183	0.3362	-0.4861
β_7	-0.3240	0.0952	-0.2514	-0.0288	0.0293
β_8	-0.1657	0.2368	-0.1679	0.7931	-0.4628
β_9	-0.2953	0.1356	0.2902	-0.2124	-0.2170
β_{10}	0.1645	0.0130	0.0367	-0.1024	0.2975
β_{11}	-0.1824	0.2022	-0.4056	0.4536	0.4061
β_{12}	-0.0551	0.0064	-0.1109	-0.1231	-0.0221
β_{13}	0.2487	-0.2126	-0.0937	-0.1759	-0.2054
β_{14}	-0.2071	0.0343	0.2623	-0.1968	-0.1943
β_{15}	0.0751	-0.0719	0.0333	-0.1157	-0.1177
β_{16}	0.1388	0.0003	0.0750	-0.5790	0.2801
β_{17}	-0.1569	0.1728	0.1653	-0.9211	0.9221
β_{18}	-0.0235	-0.0812	-0.2445	0.7464	-1.4481

can see from these two tables that the model estimation will generally give accurate results for the mean function even with relative large noise level. But for the very big noise-signal ratio, the estimates will have a large difference from the true value, such as the $2nd$ element for $\boldsymbol{\beta}_{16}$ in this simulation example.

Table 3.2. $MSE * 10^3$ of coefficient matrix estimation for simulation of multivariate kriging model

	$c = 0.01$	$c = 0.1$	$c = 0.5$	$c = 1$	$c = 2$	$c = 5$	$c = 10$
$\boldsymbol{\beta}_0$	0.0007	0.0072	0.0232	0.0374	0.0242	0.5675	0.3166
$\boldsymbol{\beta}_1$	0.0000	0.0002	0.0016	0.0013	0.0001	0.0352	0.0460
$\boldsymbol{\beta}_2$	0.0000	0.0002	0.0007	0.0014	0.0033	0.0095	0.0250
$\boldsymbol{\beta}_3$	0.0002	0.0018	0.0069	0.0101	0.0113	0.1609	0.1210
$\boldsymbol{\beta}_4$	0.0003	0.0023	0.0089	0.0146	0.0123	0.1853	0.1275
$\boldsymbol{\beta}_5$	0.0001	0.0011	0.0047	0.0086	0.0058	0.1079	0.0909
$\boldsymbol{\beta}_6$	0.0002	0.0021	0.0080	0.0138	0.0089	0.1525	0.1362
$\boldsymbol{\beta}_7$	0.0001	0.0007	0.0035	0.0050	0.0097	0.0685	0.0777
$\boldsymbol{\beta}_8$	0.0000	0.0003	0.0013	0.0017	0.0060	0.0174	0.0134
$\boldsymbol{\beta}_9$	0.0002	0.0023	0.0079	0.0137	0.0093	0.1864	0.1273
$\boldsymbol{\beta}_{10}$	0.0002	0.0015	0.0054	0.0086	0.0085	0.1368	0.0866
$\boldsymbol{\beta}_{11}$	0.0002	0.0019	0.0074	0.0125	0.0121	0.1594	0.1025
$\boldsymbol{\beta}_{12}$	0.0000	0.0005	0.0019	0.0031	0.0041	0.0345	0.0390
$\boldsymbol{\beta}_{13}$	0.0002	0.0019	0.0069	0.0111	0.0082	0.1478	0.1010
$\boldsymbol{\beta}_{14}$	0.0002	0.0018	0.0062	0.0115	0.0072	0.1444	0.0858
$\boldsymbol{\beta}_{15}$	0.0004	0.0035	0.0133	0.0221	0.0234	0.2810	0.2158
$\boldsymbol{\beta}_{16}$	0.0002	0.0022	0.0077	0.0119	0.0122	0.1734	0.1123
$\boldsymbol{\beta}_{17}$	0.0001	0.0008	0.0033	0.0066	0.0066	0.0404	0.0441
$\boldsymbol{\beta}_{18}$	0.0000	0.0002	0.0011	0.0021	0.0021	0.0137	0.0384

Since the true model is known, we can easily decompose it using functional

Table 3.3. Error percentage of coefficient matrix estimation for simulation of multivariate kriging model for $c = 10$

	1	2	3	4	5
$\boldsymbol{\beta}_0$	0.0268	0.0316	0.0367	0.0247	0.0073
$\boldsymbol{\beta}_1$	0.1222	0.0712	0.2744	0.0089	0.0188
$\boldsymbol{\beta}_2$	0.0385	0.0201	0.0131	0.0652	0.0010
$\boldsymbol{\beta}_3$	0.0263	0.2394	0.0375	0.0287	0.0024
$\boldsymbol{\beta}_4$	0.1597	0.1964	0.0530	0.0191	0.0480
$\boldsymbol{\beta}_5$	0.0802	0.0460	0.0179	0.0237	0.0379
$\boldsymbol{\beta}_6$	0.9123	0.1418	0.0376	0.0311	0.0040
$\boldsymbol{\beta}_7$	0.0276	0.0773	0.0448	0.2198	0.0403
$\boldsymbol{\beta}_8$	0.0076	0.0054	0.0119	0.0016	0.0005
$\boldsymbol{\beta}_9$	0.0528	0.0884	0.0625	0.0450	0.0082
$\boldsymbol{\beta}_{10}$	0.0635	0.6281	0.3370	0.0648	0.0042
$\boldsymbol{\beta}_{11}$	0.0672	0.0478	0.0362	0.0175	0.0036
$\boldsymbol{\beta}_{12}$	0.0850	0.5792	0.0504	0.0244	0.0253
$\boldsymbol{\beta}_{13}$	0.0490	0.0448	0.1542	0.0439	0.0070
$\boldsymbol{\beta}_{14}$	0.0509	0.2357	0.0466	0.0326	0.0062
$\boldsymbol{\beta}_{15}$	0.3427	0.2832	0.9302	0.1451	0.0266
$\boldsymbol{\beta}_{16}$	0.0976	33.4259	0.2141	0.0149	0.0057
$\boldsymbol{\beta}_{17}$	0.0328	0.0241	0.0385	0.0038	0.0007
$\boldsymbol{\beta}_{18}$	0.1904	0.0446	0.0226	0.0041	0.0004

ANOVA. For example, the total variance for the j^{th} component is

$$\begin{aligned} D^j = \text{Var}g^j \quad = \quad & \tfrac{1}{12}\sum_{i=1}^{18}\beta_{ij} + a_j^2c^2\{\int exp(-\mathbf{x}\mathbf{A}_j\mathbf{x}')d\mathbf{x} - [\int exp(-\tfrac{1}{2}\mathbf{x}\mathbf{A}_j\mathbf{x}')d\mathbf{x}]^2\} \\ & +2a_jc\{\int \boldsymbol{\beta}_{.j}\mathbf{x}exp(-\tfrac{1}{2}\mathbf{x}\mathbf{A}_j\mathbf{x}')d\mathbf{x} - (\beta_{0j} + \tfrac{1}{2}\sum_{i=1}^{18}\beta_{ij})exp(-\tfrac{1}{2}\mathbf{x}\mathbf{A}_j\mathbf{x}')d\mathbf{x}\}, \end{aligned} \tag{3.73}$$

where $a_j = \sin(2k\pi t_j) + \cos(2k\pi t_j)$. Using Monte-Carlo integration, the calculation for these decomposed items are straightforward. Then we calculate the functional ANOVA decomposition by modeling it using Gaussian kriging model. The true overall variance and the difference from the simulation result for the case $c = 10$ is given in table 3.4. We can see the calculation for the overall variance is accurate. The true value of Var(g_i^j) (the variance of linear effect terms, where $i = 1, \cdots, 18$ and $j = 1, \cdots, 5$) for $c = 5$ are given in table 3.5. The corresponding simulation error rate is given in table 3.6. We can see from these tables that the simulation result for these variance of functional ANOVA decomposed linear effect terms is accurate even for small value of variance terms.

Table 3.4. Total variance and its estimate difference for functional ANOVA decomposition by multivariate kriging model for $c = 10$

	D^1	D^2	D^3	D^4	D^5
True	0.06421	0.02331	0.14464	0.28954	0.38014
Difference*10^4	-0.7614	0.5309	0.3288	-0.2864	0.0378

Table 3.7 gives the error rate for the variance of decomposed linear terms for $c = 10$. For this noise level, we can see the error rate for smaller value of variance term is large.

Finally, the sum of Sobol's indices for linear effect term is calculated and compared with the simulation result. For the ith decomposed linear term in the jth component, g_i^j, where $i = 1, \cdots, 18$, and $j = 1, \cdots, 5$, the Sobol's indices is:

$$s_i^j = \frac{\text{Var}g_i^j}{\text{Var}g^j}. \tag{3.74}$$

Table 3.5. True variance of linear effect terms for functional ANOVA decomposition by multivariate kriging model for $c = 5$

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
$i = 1$	0.0002	0.0003	0.0000	0.0138	0.0001
$i = 2$	0.0004	0.0011	0.0064	0.0001	0.0126
$i = 3$	0.0246	0.0002	0.0177	0.0089	0.0440
$i = 4$	0.0008	0.0003	0.0098	0.0222	0.0001
$i = 5$	0.0015	0.0029	0.0437	0.0071	0.0001
$i = 6$	0.0000	0.0007	0.0222	0.0093	0.0195
$i = 7$	0.0087	0.0007	0.0052	0.0001	0.0001
$i = 8$	0.0023	0.0046	0.0023	0.0519	0.0177
$i = 9$	0.0072	0.0015	0.0070	0.0037	0.0039
$i = 10$	0.0022	0.0000	0.0001	0.0009	0.0073
$i = 11$	0.0027	0.0034	0.0136	0.0170	0.0136
$i = 12$	0.0003	0.0000	0.0010	0.0012	0.0000
$i = 13$	0.0051	0.0037	0.0007	0.0026	0.0035
$i = 14$	0.0035	0.0001	0.0057	0.0032	0.0031
$i = 15$	0.0005	0.0004	0.0001	0.0011	0.0011
$i = 16$	0.0016	0.0000	0.0005	0.0277	0.0065
$i = 17$	0.0020	0.0025	0.0023	0.0700	0.0702
$i = 18$	0.0000	0.0005	0.0049	0.0460	0.1730

Table 3.6. Error rate for linear term variance for functional ANOVA decomposition by multivariate kriging model for $c = 5$

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
$i = 1$	0.0257	0.0088	0.0080	0.0100	0.0098
$i = 2$	0.0168	0.0087	0.0105	0.0080	0.0098
$i = 3$	0.0094	0.0120	0.0099	0.0097	0.0098
$i = 4$	0.0140	0.0075	0.0100	0.0099	0.0098
$i = 5$	0.0163	0.0083	0.0094	0.0101	0.0097
$i = 6$	0.0147	0.0087	0.0097	0.0097	0.0098
$i = 7$	0.0078	0.0070	0.0088	0.0066	0.0097
$i = 8$	0.0071	0.0091	0.0087	0.0099	0.0098
$i = 9$	0.0109	0.0107	0.0097	0.0099	0.0098
$i = 10$	0.0093	0.0168	0.0097	0.0099	0.0098
$i = 11$	0.0039	0.0079	0.0090	0.0101	0.0098
$i = 12$	0.0036	0.0199	0.0080	0.0093	0.0099
$i = 13$	0.0091	0.0094	0.0098	0.0099	0.0098
$i = 14$	0.0108	0.0114	0.0100	0.0098	0.0098
$i = 15$	0.0006	0.0058	0.0052	0.0104	0.0098
$i = 16$	0.0080	0.7914	0.0094	0.0098	0.0098
$i = 17$	0.0063	0.0087	0.0110	0.0097	0.0098
$i = 18$	0.0072	0.0118	0.0091	0.0099	0.0098

Table 3.7. Error rate for linear term variance for functional ANOVA decomposition by multivariate kriging model for $c = 10$

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
$i = 1$	0.6187	0.3499	0.8862	0.0170	0.0415
$i = 2$	0.1451	0.0461	0.0229	0.2531	0.0097
$i = 3$	0.0099	0.0248	0.0100	0.0101	0.0098
$i = 4$	0.0889	0.1186	0.0184	0.0108	0.0167
$i = 5$	0.0175	0.0133	0.0107	0.0093	0.0125
$i = 6$	0.6369	0.0484	0.0122	0.0119	0.0098
$i = 7$	0.0120	0.0360	0.0191	0.1610	0.0170
$i = 8$	0.0116	0.0116	0.0158	0.0097	0.0097
$i = 9$	0.0121	0.0150	0.0131	0.0110	0.0099
$i = 10$	0.0171	0.4113	0.1685	0.0162	0.0098
$i = 11$	0.0073	0.0118	0.0110	0.0092	0.0099
$i = 12$	0.3291	0.9591	0.1561	0.0494	0.0493
$i = 13$	0.0108	0.0105	0.0190	0.0106	0.0098
$i = 14$	0.0119	0.0578	0.0119	0.0107	0.0098
$i = 15$	0.0111	0.0154	0.0520	0.0093	0.0101
$i = 16$	0.0276	0.9998	0.0788	0.0104	0.0100
$i = 17$	0.0246	0.0201	0.0291	0.0102	0.0099
$i = 18$	0.1818	0.0179	0.0143	0.0096	0.0098

The sum of these Sobol's indices is

$$s^j = \sum_{i=1}^{18} s_i^j = \frac{\sum_{i=1}^{18} \text{Var} g_i^j}{\text{Var} g^j}. \tag{3.75}$$

Table 3.8 shows the true and simulated result comparison for s^j for $c = 10$. From this table, we can see the simulation result is very close to the true value, and the decomposed linear term explains more than 99 percent of the total variance for each component. From these simulation result, our simulation code is proven to be accurate and trustable.

Table 3.8. Sum of Sobol's indices for linear terms of functional ANOVA decomposition by multivariate kriging model for $c = 10$

	s^1	s^2	s^3	s^4	s^5
True	0.9902	0.9903	0.9902	0.9902	0.9902
Simulation	0.9914	0.9880	0.9899	0.9903	0.9902

3.6 Comparison: Multivariate Kriging v.s. Kriging with single variate response

In this section, we compare the two kriging models: multivariate kriging which is proposed in previous sections, and kriging with single variate response.

We use the same simulation data set which is generated in section 3.5. To compare these two versions of kriging models, we use leave-one-out cross validation method to calculate the prediction error for each model. In the simulation formula (3.72), the predictor variables are $x_{i,k}$, where $i = 1, 2, \cdots, 30$, $k = 1, 2, \cdots, 18$. The response variable is $y_{i,j}$, where $i = 1, 2, \cdots, 30$, $j = 1, 2, 3, 4, 5$.

For leave-one-out prediction, we remove each of 30 design cases from both design matrix and response variables in sequence. Denote

$$\mathbf{Y}^{(-i)} = \begin{pmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,5} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,5} \\ \vdots & \vdots & \vdots & \vdots \\ y_{i-1,1} & y_{i-1,2} & \cdots & y_{i-1,5} \\ y_{i+1,1} & y_{i+1,2} & \cdots & y_{i+1,5} \\ \vdots & \vdots & \vdots & \vdots \\ y_{30,1} & y_{30,2} & \cdots & y_{30,5} \end{pmatrix},$$

$$\mathbf{y}_j^{(-i)} = (y_{1,j}, y_{2,j}, \cdots, y_{i-1,j}, y_{i+1,j}, \cdots, y_{30,j})^T,$$

$$\mathbf{y}^i = (y_{i,1}, y_{i,2}, \cdots, y_{i,5}),$$

$$\mathbf{X}^{(-i)} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,18} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,18} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i-1,1} & x_{i-1,2} & \cdots & x_{i-1,18} \\ x_{i+1,1} & x_{i+1,2} & \cdots & x_{i+1,18} \\ \vdots & \vdots & \vdots & \vdots \\ x_{30,1} & x_{30,2} & \cdots & x_{30,18} \end{pmatrix},$$

$$\mathbf{x}^i = (x_{i,1}, x_{i,2}, \cdots, x_{i,18}).$$

And

$$P.E.1 = \sum_{i=1}^{30} (\mathbf{y}^i - \hat{\mathbf{y}}^i)^2$$

for prediction error for multivariate kriging model, where $\hat{\mathbf{y}}^i$ is the prediction from multivariate kriging model for leave the ith observation out at $\mathbf{x} = \mathbf{x}^i$;

$$P.E.2 = \sum_{j=1}^{5} \sum_{i=1}^{30} (y_j^i - \hat{y}_j^i)^2$$

for prediction error from kriging model with single response, where $\hat{y}_j^i$ is the prediction for leave the ith observation out at $\mathbf{x} = \mathbf{x}^i$ for the jth component.

Table 3.9 shows the average of prediction error for 5000 simulation using formula (3.72), where gain is defined as the ratio of the difference of two prediction error and the prediction error of the kriging with single response. From it, we can see that both kriging model did good job. For the data which is generated with small noise level, the two model can not make any difference from the prediction error point of view. But when the noise level becomes larger, multivariate kriging model become more and more powerful than the kriging model with single response. From this simulation study, we can see the advantages of multivariate kriging model over the naive way.

Table 3.9. Prediction error comparison for multivariate kriging model and kriging with single response

	0.5	1	2	5	10
P.E.(multivariate)	2.7e-07	8.4e-06	4.2e-05	3.2e-05	1.0e-05
P.E.(single)	2.9e-07	8.6e-06	4.7e-05	3.9e-05	1.5e-05
Gain	0.0741	0.0238	0.1064	0.1795	0.3333

3.7 Case Study

In this section, we apply the proposed methodology for a thorough analysis of computer simulation data in the design of engine structure to minimize radiated noise (Example 1.1). This data set has been analyzed in Chapter 7 of Fang, et al. (2005). To optimize the design of the cylinder block, 17 design variables listed in Table 1.1 were chosen. The variable *Dam* has three levels, and therefore there are 18 x-variables. A uniform design with 30 runs, given in Table 7.2 of Fang, et al. (2005), was used for the experiment. In this example, the output is the structure response (z) of the oil pan flange in terms of acceleration at 6 engine RPMs, 1000, 2000, ..., 6000 RPM, which directly influences the radiated noise through the vibro-acoustic relationship. The outputs are presented in Table 7.3 of Fang, et al. (2005), in which the response were treated as functional response with sparse sample rate. Here we will treat it as a multiple response. It is known that the response is exponentially increasing as the level RPM increases and typically is modeled by

$$z = \exp(\alpha_0 + \alpha_1 \text{RPM}). \tag{3.76}$$

To understand the impact of RPM on the structure response, we are interested in the impact of design variables on the parameter α_1. Thus, we take the response y to be difference of the natural logarithm of the noise value between the adjacent RPM levels. That is, for $j = 1, \cdots, 5$,

$$y_j = \log(z_{j+1}) - \log(z_j) = 1000 * \alpha_1,$$

where z_j is the structure response at $j \times 1000$ RPM. This response value is plotted in Figure 1.2. We can see the lines are almost parallel to each other for different observations and there is a slight curvature along RPM direction. We will denote these responses as $\mathbf{Y} = (\mathbf{Y}_1^T, \cdots, \mathbf{Y}_{30}^T)^T$. Note that for each observation, the response $\mathbf{Y}_i$ is a 5-component vector.

Other than the shape of the curves, the most significant fact in this plot is that the variation caused by different design cases is significantly less than the variation caused by different RPM levels. But RPM level is not a control factor. For this simulation study, engineers want to understand which design variable is the most impacting factor and also want to know how each variable affects the response under different RPM levels. They want to lower the engine noise to the minimum through the control of the design variables under different RPM level. Hence, the thorough sensitivity analysis for the engine noise with respect to each design variable is required although the variation caused by this is not even on the same scale as the variation caused by RPM level. In the mean time, engineers would also like to get a quick prediction at undesigned point.

We will use partial sum of squares in regression to do the sensitivity analysis first. This approach allows researchers to get a rough idea of the importance rank for each variable including the RPM level, design variable and all the interaction terms. We will also get the regression coefficients for the corresponding selected terms. From the sign of these coefficients, we can also get a rough idea how each item affects the output. Then we will develop several versions of multivariate kriging for this case so that our models are guaranteed to interpolate the observations. Then we will use functional ANOVA decomposition to get the marginal effect for each design variable. The result from the functional ANOVA can be viewed as the result which already exclude all the other variables effect hence it is more accurate to reflect the effect of this variable.

3.7.1 Sensitivity Analysis using Regression Partial Sum of Squares

We start with sensitivity analysis by using the partial sum of squares. We can view the response under different designs with RPM level as single value responses with the 18 variables plus the 5 different RPM difference levels as the predictor. Then we can do a linear regression and use partial sum of square as the criteria for measuring the importance of factors in the model. The goal of this section is to use partial sum of square to select the most significant/sensitive predictors among all linear and second order interaction terms.

Define the extra sum of squares of $\mathbf{x}_{(2)}$ given $\mathbf{x}_{(1)}$ as

$$SSR(\mathbf{x}_{(2)}|\mathbf{x}_{(1)}) = SSR(\mathbf{x}_{(2)}, \mathbf{x}_{(1)}) - SSR(\mathbf{x}_{(1)}), \tag{3.77}$$

where $\mathbf{x}_{(1)}$ is a sub-vector of predictor set $\mathbf{x}$, $\mathbf{x}_{(2)}$ is another sub-vector of predictor set which components are not in $\mathbf{x}_{(1)}$, and $SSR(\mathbf{x}_{(1)})$ is the regression sum of squares when the regression model only includes the predictor set $\mathbf{x}_{(1)}$. This extra sum of squares measures the increase of regression sum of squares for adding the predictor set $\mathbf{x}_{(2)}$ into the original predictor set $\mathbf{x}_{(1)}$. If this value is large, it shows that $\mathbf{x}_{(2)}$ is a relative important/sensitive predictor set.

In this case, since Figure 1.2 shows a slight interaction shape with almost parallel lines, we let total predictor set $\mathbf{x}$ includes the RPM difference level, all the 18 linear terms in table 1.2, and all the second order interaction terms associated with these terms. The regression model is

$$y = \beta_0 x_0 + \sum_{i=1}^{19} \beta_i x_i + \sum_{i=1}^{19} \gamma_i x_i^2 + \sum_{i<j} \tau_{ij} x_i x_j + \epsilon, \tag{3.78}$$

where $x_0 = 1$ corresponding to the intercept, $x_1, \cdots, x_{18}$ are defined in table 1.2, $x_{19} = 0, 1, \cdots, 4$ is the RPM difference level. Since we put x_{19} as one of the predictor, the total observation number for this linear regression is $30 \times 5 = 150$.

The first step is to calculate $SSR(x_0)$, the regression sum of squares with only the intercept. The result is: $SSR(x_0) = 155.4366$ and the corresponding $\beta_0 = 1.10180$. Along with these SSRs and the corresponding coefficient estimates, we also calculated the prediction errors in the form of sum of square error: $P.E. = \sum_{i=1}^{n}(Y_i - \hat{y}_i)^2$. For the model with only intercept, $P.E. = 1905.3$.

Next, we begin the variable screening within the linear effect terms $x_1, \cdots, x_{19}$. The extra sum of squares for the main effects $SSR(\cdot|x_0)$ is given in tables 3.10 and 3.11. Not surprisingly, x_{19} (RPM difference level) has the largest value hence x_{19} is the most important linear effect. The estimated coefficient is -0.2515, which is given in Table 3.12. The negative sign shows that the difference of the $\log$ engine noise will become smaller with higher RPM difference level. But keep in mind that the RPM variable is not a control variable and our goal is to adjust the most significant design variables so that we can adjust the engine noise under different RPM value. Hence

Table 3.10. Extra Sum of Squares for main effects

$SS * 100$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	P.E.
$SSR(\cdot\|x_0)$	.2293	.2622	.2828	.1827	.0303	.7238	.0008	.0003	.3278	1905.3
$SSR(\cdot\|x_{(1)})$	.2293	.2622	.2828	.1827	.0303	.7238	.0008	.0003	.3278	491.9
$SSR(\cdot\|x_{(2)})$	.2818	.0118	.3528	.0956	.0712	**.4690**	.0300	.0014	.1015	490.2
$SSR(\cdot\|x_{(3)})$	.2242	.0220	**.4394**	.0688	.0198	-	.0223	.0000	.0539	489.5
$SSR(\cdot\|x_{(4)})$	.2106	.0488	-	.1201	.0152	-	.1345	.0121	.0690	488.9
$SSR(\cdot\|x'_{(5)})$	.0772	.0203	-	**.1737**	.0061	-	.0250	.0000	.0791	488.2
$SSR(\cdot\|x'_{(6)})$	.0898	.0184	-	-	.0132	-	.0450	.0058	**.1334**	488.0

Table 3.11. Extra Sum of Squares for main effect(continued)

SS*100	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
$SSR(\cdot\|x_0)$	.0156	.0534	.0377	.0008	.0544	.0100	.1183	.0001	1.1579	**942.1968**
$SSR(\cdot\|x_{(1)})$	.0156	.0534	.0377	.0008	.0544	.0100	.1183	.0001	**1.1579**	-
$SSR(\cdot\|x_{(2)})$	.0014	.1700	.0004	.0135	.1038	.0000	.0633	.0240	-	-
$SSR(\cdot\|x_{(3)})$	.0177	.2042	.0636	.0183	.1268	.0081	.0027	.0270	-	-
$SSR(\cdot\|x_{(4)})$	.0009	**.3604**	.0187	.0294	.1099	.0002	.0066	.0910	-	-
$SSR(\cdot\|x'_{(5)})$	.0014	-	.0040	.0531	.1442	.0004	.0257	.1184	-	-
$SSR(\cdot\|x'_{(6)})$	.0257	-	.0301	.1096	.1183	.0073	.0176	.0766	-	-

Table 3.12. Selected variables

$x_{(1)}$	(x_0, x_{19})
$\beta_{(1)}$	$(1.0180, -0.2515)$
$x_{(2)}$	(x_0, x_{19}, x_{18})
$\beta_{(2)}$	$(1.0180, -0.2515, -0.0088)$
$x_{(3)}$	$(x_0, x_{19}, x_{18}, x_6)$
$\beta_{(3)}$	$(1.0180, -0.2515, -0.0079, -0.0057)$
$x_{(4)}$	$(x_0, x_{19}, x_{18}, x_6, x_3)$
$\beta_{(4)}$	$(1.0180, -0.2515, -0.0081, -0.0061, 0.0055)$
$x'_{(5)}$	$(x_0, x_{19}, x_{18}, x_6, x_3, x_{11})$
$\beta'_{(5)}$	$(1.0180, -0.2515, -0.0089, -0.0066, 0.0065, 0.0051)$
$x'_{(6)}$	$(x_0, x_{19}, x_{18}, x_6, x_3, x_{11}, x_5)$
$\beta'_{(6)}$	$(1.0180, -0.2515, -0.0087, -0.0064, 0.0070, 0.0055, -0.0035)$
$x_{(5)}$	$(x_0, x_{19}, x_{18}, x_6, x_3, x_{19}^2)$
$\beta_{(5)}$	$(1.0438, -0.2515, -0.0081, -0.0062, 0.0055, -0.0260)$
$x_{(6)}$	$(x_0, x_{19}, x_{18}, x_6, x_3, x_{19}^2, x_{18}x_{19})$
$\beta_{(6)}$	$(1.0438, -0.2515, -0.0081, -0.0062, 0.0055, -0.0260, -0.0085)$
$x_{(7)}$	$(x_0, x_{19}, x_{18}, x_6, x_3, x_{19}^2, x_{18}x_{19}, x_1x_{19})$
$\beta_{(7)}$	$(1.0438, -0.2515, -0.0081, -0.0062, 0.0055, -0.0260, -0.0088, 0.0069)$
$x_{(8)}$	$(x_0, x_{19}, x_{18}, x_6, x_3, x_{19}^2, x_{18}x_{19}, x_1x_{19}, x_{13}x_{19})$
$\beta_{(8)}$	$(1.0438, -0.2515, -0.0081, -0.0062, 0.0055, -0.0260, -0.0093, 0.0079, -0.0060)$

Table 3.13. Selected variables–continued

	$\beta_{(9)}$	$\beta_{(10)}$	$\beta_{(11)}$	$\beta_{(12)}$	$\beta_{(13)}$	$\beta_{(14)}$
x_0	1.0436	1.0436	1.0436	1.0436	1.0436	1.0436
x_{19}	-0.2515	-0.2515	-0.2515	-0.2515	-0.2515	-0.2515
x_{18}	-0.0078	-0.0078	-0.0078	-0.0080	-0.0080	-0.0080
x_6	-0.0053	-0.0053	-0.0053	-0.0054	-0.0054	-0.0054
x_3	0.0046	0.0046	0.0046	0.0050	0.0050	0.0050
x_{19}^2	-0.0260	-0.0260	-0.0260	-0.0260	-0.0260	-0.0260
$x_{18}x_{19}$	-0.0093	-0.0101	-0.0109	-0.0109	-0.0131	-0.0135
x_1x_{19}	0.0079	0.0072	0.0055	0.0055	0.0037	0.0034
$x_{13}x_{19}$	-0.0060	-0.0055	-0.0056	-0.0056	-0.0052	-0.0053
x_1x_{15}	-0.0071	-0.0071	-0.0071	-0.0079	-0.0079	-0.0079
$x_{17}x_{19}$	—	0.0054	0.0057	0.0057	0.0061	0.0068
$x_{11}x_{19}$	—	—	0.0054	0.0054	0.0067	0.0076
$x_{11}x_{17}$	—	—	—	0.0049	0.0049	0.0049
x_2x_{19}	—	—	—	—	-0.0055	-0.0053
x_3x_{19}	—	—	—	—	—	0.0037

more importantly, we need to study the sensitivity of design variables and probably the interactions between them. We then calculate $SSR(\cdot|x_0, x_{19})$ with result given on the second line in tables 3.10 and 3.11. This time, the most important linear effect is x_{18} and the estimated coefficient is -0.0088 (in Table 3.12). The difference of log engine noise has the same tendency with respect to this variable as it has with respect to RPM difference level. We continue to repeat these steps, and finally we reach $x'_{(6)}$. We can see that after the big jump on $x_{(4)}$, the decreasing of extra sum of squares is flat. So in the main effect terms, we will pick the subset $x_{(4)} = (x_0, x_{19}, x_{18}, x_6, x_3)$. The definition of each selected $x_{(\cdot)}$ are given in table 3.12 along with the corresponding regression coefficient. Note that different from the other three variables, the difference of log engine noise tends to increase along with the increase of x_3.

Next, we proceed to the second order interaction terms. Tables 3.15 and 3.16 give the extra sum of squares for each second order interaction term $SSR(\cdot|x_{(4)})$. Again, not surprisingly, the term x_{19}^2 has the largest value, which shows that x_{19}^2 is the most important second order term. Similarly, Tables 3.15 and 3.16 give $SSR(\cdot|x_{(5)})$. We can see this time, the term $x_{18}x_{19}$ is the most important second order term. Table 3.14 gives the sequential maximum extra sum of squares for the second derivative terms along with the prediction error. Due to the flat shape on the last two selection steps, the final selected predictor subset is $x_{(12)}$. The corresponding estimated coefficients are given in Tables 3.12 and 3.13.

Table 3.14. Maximum extra sum of squares for second order interaction terms

	Max $SSR(\cdot\|x_{(i)})$	P.E.
$x_{(4)}$	7.0256	488.90
$x_{(5)}$	1.0573	478.38
$x_{(6)}$	0.7105	476.82
$x_{(7)}$	0.5125	475.77
$x_{(8)}$	0.6937	475.02
$x_{(9)}$	0.4094	474.01
$x_{(10)}$	0.3813	473.41
$x_{(11)}$	0.3409	472.87
$x_{(12)}$	0.3252	472.38
$x_{(13)}$	0.1796	471.92
$x_{(14)}$	0.2145	471.68

From this extra sum of square analysis, we know that:

- The most important factors are x_{19} and x_{19}^2, but they are not control variables.

- Along with the increase of x_{19} and x_{19}^2, the difference of log engine noise is decreasing.
- It is desirable that the difference of log engine noise decreases.
- From the analysis result, we know that the difference of log engine noise will increase with the increase of x_3, x_1x_{19}, $x_{17}x_{19}$, $x_{11}x_{19}$, $x_{11}x_{17}$.
- Hence,in the practice, we need to minimize the design variables x_3, x_1, x_{17}, x_{11}. Also, we need to maximize x_{18}, x_6, x_{13}, and x_1. Note that some of the variables are controlled through the interaction term with the RPM difference level. By doing so, we can minimize the difference of the log engine noise.
- The rough importance rank from regression partial sum squares point of view is:

$$x_0, x_{19}, x_{18}, x_6, x_3, x_{19}^2, x_{18}x_{19}, x_1x_{19}, x_{13}x_{19}, x_1x_{15}, x_{17}x_{19}, x_{11}x_{19}, x_{11}x_{17}.$$

- In order to carefully study the effect of other design factors, it is important that we treat the RPM difference term separately.

Table 3.15. Extra Sum of Squares for interaction $SSR(\cdot|x_{(4)})$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_1	.0266	-	-	-	-	-	-	-	-
x_2	.0009	.0725	-	-	-	-	-	-	-
x_3	.0841	.0703	.0170	-	-	-	-	-	-
x_4	.0004	.0587	.0026	.0880	-	-	-	-	-
x_5	.0073	.0108	.0075	.1354	.0561	-	-	-	-
x_6	.0697	.0235	.0548	.0095	.0141	.0605	-	-	-
x_7	.0663	.2540	.0005	.0217	.0267	.0485	.0573	-	-
x_8	.2964	.0019	.0915	.0009	.0199	.0415	.0797	.0347	-
x_9	.1605	.0009	.0761	.0648	.0201	.0074	.0293	.1725	-
x_{10}	.2915	.0025	.0122	.0157	.0098	.1157	.0003	.1136	.0044
x_{11}	.0005	.1444	.1397	.0263	.0185	.1199	.1004	.0854	.0914
x_{12}	.2144	.0159	.0030	.0977	.0012	.1271	.0201	.0023	.0005
x_{13}	.0997	.3028	.2858	.0797	.0063	.1250	.3167	.0022	.0073
x_{14}	.0516	.0159	.0031	.1819	.0051	.0927	.0015	.0004	.0635
x_{15}	.4952	.2426	.0117	.0025	.0348	.0217	.1052	.0283	.1162
x_{16}	.0041	.0968	.1009	.0732	.0102	.0009	.1714	.0259	.0065
x_{17}	.1139	.0921	.0955	.0594	.1071	.0420	.1939	.0571	.0975
x_{18}	.2820	.0140	.0011	.0691	.0493	.0420	.0795	.0620	.0074
x_{19}	.6285	.0341	.0060	.0122	.1471	.3174	.0620	.0998	.1063

Table 3.16. Extra Sum of Squares for interaction $SSR(\cdot|x_{(4)})$ (continued)

	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{11}	.3003	-	-	-	-	-	-	-	-	-
x_{12}	.0003	.0753	-	-	-	-	-	-	-	-
x_{13}	.0043	.0156	.0596	-	-	-	-	-	-	-
x_{14}	.0198	.0567	.0670	.0113	-	-	-	-	-	-
x_{15}	.0208	.0403	.0322	.0024	.0056	.0004	-	-	-	-
x_{16}	.1106	.0141	.0259	.0186	.2507	.0181	.0257	-	-	-
x_{17}	.0219	.2115	.2175	.0036	.1449	.0166	.0159	.0072	-	-
x_{18}	.0070	.0586	.0442	.0091	.1153	.0022	.0004	.2799	.0004	-
x_{19}	.0266	.3673	.0155	.2354	.0096	.0462	.1418	.4059	1.0573	**7.0260**

3.7.2 Multivariate Kriging Model

We need models for computer experiment interpolate the observed data. Hence we will use the multivariate version of the kriging model for this case. The multivariate version is ideal here because we want to separate the RPM effect from other predictor effects. We also want smoothness for the prediction surfaces over these 18 design variables hence we will use Gaussian correlation function in the correlation matrix in kriging model.

The Gaussian correlation function assumes the location-free and direction-free property for the correlation coefficient between any two point. The Gaussian correlation function for the correlation coefficient between any two points $\mathbf{x}_i = (x_{i,1}, \cdots, x_{i,s})$ and $\mathbf{x}_j = (x_{j,1}, \cdots, x_{j,s})$ is:

$$r(\mathbf{x}_i, \mathbf{x}_j) = exp(-\sum_{k=1}^{s} \theta_k (x_{i,k} - x_{j,k})^2) \tag{3.79}$$

Table 3.17. Extra Sum of Squares for interaction $SSR(\cdot|x_{(5)})$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_1	.0266	-	-	-	-	-	-	-	-
x_2	.0009	.0725	-	-	-	-	-	-	-
x_3	.0841	.0703	.0170	-	-	-	-	-	-
x_4	.0004	.0587	.0026	.0880	-	-	-	-	-
x_5	.0073	.0108	.0075	.1354	.0561	-	-	-	-
x_6	.0697	.0235	.0548	.0095	.0141	.0605	-	-	-
x_7	.0663	.2540	.0005	.0217	.0267	.0485	.0573	-	-
x_8	.2964	.0019	.0915	.0009	.0199	.0415	.0797	.0347	-
x_9	.1605	.0009	.0761	.0648	.0201	.0074	.0293	.1725	-
x_{10}	.2915	.0025	.0122	.0157	.0098	.1157	.0003	.1136	.0044
x_{11}	.0005	.1444	.1397	.0263	.0185	.1199	.1004	.0854	.0914
x_{12}	.2144	.0159	.0030	.0977	.0012	.1271	.0201	.0023	.0005
x_{13}	.0997	.3028	.2858	.0797	.0063	.1250	.3167	.0022	.0073
x_{14}	.0516	.0159	.0031	.1819	.0051	.0927	.0015	.0004	.0635
x_{15}	.4952	.2426	.0117	.0025	.0348	.0217	.1052	.0283	.1162
x_{16}	.0041	.0968	.1009	.0732	.0102	.0009	.1714	.0259	.0065
x_{17}	.1139	.0921	.0955	.0594	.1071	.0420	.1939	.0571	.0975
x_{18}	.2820	.0140	.0011	.0691	.0493	.0420	.0795	.0620	.0074
x_{19}	.6285	.0341	.0060	.0122	.1471	.3174	.0620	.0998	.1063

Table 3.18. Extra Sum of Squares for interaction $SSR(\cdot|x_{(5)})$ (continued)

	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{11}	.3003	-	-	-	-	-	-	-	-	-
x_{12}	.0003	.0753	-	-	-	-	-	-	-	-
x_{13}	.0043	.0156	.0596	-	-	-	-	-	-	-
x_{14}	.0198	.0567	.0670	.0113	-	-	-	-	-	-
x_{15}	.0208	.0403	.0322	.0024	.0056	-	-	-	-	-
x_{16}	.1106	.0141	.0259	.0186	.2507	.0181	-	-	-	-
x_{17}	.0219	.2115	.2175	.0036	.1449	.0166	.0159	-	-	-
x_{18}	.0070	.0586	.0442	.0091	.1153	.0022	.0004	.2799	0.0004	-
x_{19}	.0266	.3673	.0155	.2354	.0096	.0462	.1418	.4059	**1.0573**	-

for the multiple variable version, where $\theta_k > 0$, and:

$$r(\mathbf{x}_i, \mathbf{x}_j) = exp(-\theta \sum_{k=1}^{s} (x_{i,k} - x_{j,k})^2) \tag{3.80}$$

for the single variable version, where $\theta > 0$. The advantage for the multiple variables is its flexibility to model the data, but at the same time, it increases the complexities of the problem. For the Fisher scoring algorithm, we need to supply good initial values for these unknown parameters. When the number of unknown parameters is large, it will be more and more difficult to find good initial values. The single variable Gaussian correlation function will greatly simplify the estimation procedure and furthermore, it will allow us to use cross-validation procedures to estimate the unknown parameter. However it will lack the flexibility to model the different degree of correlation in different component direction.

MLE for Multiple Variable Gaussian Correlation Function

We start by using (3.79) and the Fisher scoring algorithm to construct the multiple variable kriging model for this data. The kriging model is

$$\mathbf{y}(\mathbf{x}) = B_0 x_0 + \sum_{i=1}^{18} B_i x_i + \mathbf{r}(\mathbf{x})\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{X}B)$$

where $x_0 = 1$, $x_1, \cdots, x_{18}$ are 18 predictors listed in Table 1.1. Here, $B_0, \cdots, B_{18}$ are vectors with length 5, which is the same with $\mathbf{y}$. The design matrix $\mathbf{X}$ is composed of the intercept column together with 18 design variables listed in Tables 1.2 and 1.3. Hence we consider the base/intercept effect together with the main effect. We use (3.79) to construct $\mathbf{R}$ and $\mathbf{r}_0$ in the kriging model (3.31) for a given $\boldsymbol{\theta}$. We use (3.37) to estimate B and use Fisher scoring algorithm to estimate $\boldsymbol{\theta}$.

We experienced the difficulty of a near singular Hessian matrix in Fisher scoring algorithm for this dataset. In addition to this, we found that estimation for B and Ω are not sensitive to the initial $\boldsymbol{\theta}$ value, but the estimate for $\boldsymbol{\theta}$ will change with different initial value. Furthermore, the likelihood value itself is not sensitive to the initial value. A typical estimated B is given below together with the estimated Ω which is given in Table 3.19. The corresponding estimated $\boldsymbol{\theta}$ is also given below.

$$\hat{\Psi} = \begin{pmatrix} 0.0013 & -0.0005 & -0.0006 & 0.0004 & 0.0001 \\ -0.0005 & 0.0006 & -0.0002 & 0.0001 & 0.0004 \\ -0.0006 & -0.0002 & 0.0031 & -0.0027 & 0.0005 \\ 0.0004 & 0.0001 & -0.0027 & 0.0054 & -0.0022 \\ 0.0001 & 0.0004 & 0.0005 & -0.0022 & 0.0056 \end{pmatrix}, \qquad (3.81)$$

and

$$\begin{array}{l} (\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_5, \hat{\theta}_6, \hat{\theta}_8, \hat{\theta}_9, \hat{\theta}_{13}, \hat{\theta}_{17}, \hat{\theta}_{18}) = \\ (0.0844, 0.0974, 0.0333, 0.0250, 0.1461, 0.0431, 0.1757, 0.8250, 0.6341) \end{array}, \qquad (3.82)$$

where all other components are zero.

Table 3.19. $\hat{B}$ for kriging model using (3.79)

$\hat{B}$	1	2	3	4	5
1	**1.4483**	**0.9821**	**1.1112**	**0.9899**	**0.5584**
2	0.0008	0.0004	-0.0033	0.0184	-0.0041
3	0.0035	0.0041	0.0093	-0.0010	-0.0150
4	0.0197	0.0017	-0.0183	0.0145	0.0266
5	-0.0012	-0.0036	-0.0142	-0.0183	-0.0013
6	0.0071	0.0073	-0.0311	0.0123	0.0011
7	0.0001	-0.0036	-0.0183	0.0104	-0.0187
8	-0.0132	0.0042	-0.0084	-0.0044	0.0017
9	-0.0080	0.0086	-0.0062	0.0291	-0.0142
10	-0.0140	0.0067	0.0115	-0.0074	-0.0103
11	0.0078	0.0005	0.0018	-0.0029	0.0142
12	-0.0111	0.0111	-0.0173	0.0202	0.0232
13	-0.0017	-0.0007	-0.0072	-0.0047	0.0008
14	0.0127	-0.0106	-0.0046	-0.0086	-0.0100
15	-0.0100	0.0010	0.0120	-0.0096	-0.0097
16	0.0050	-0.0041	0.0005	-0.0057	-0.0070
17	0.0083	-0.0005	0.0011	-0.0260	0.0114
18	-0.0073	0.0071	0.0051	-0.0369	0.0399
19	-0.0012	-0.0027	-0.0072	0.0262	-0.0629

From above results, we noticed that:

- Compared to the intercept term, the linear effect for all the predictors is very small. This suggest us that we may do not need to put the 18 design variables in the design matrix.
- There is a strong correlations between different RPM levels. For example, in (3.81), the correlation coefficient between the first and second component are: $\rho_{21} = \frac{-0.0005}{\sqrt{0.0015 \cdot 0.0007}} = -0.488$. So we can not simply apply kriging model to each RPM difference level using the univariate version.
- The estimation is not stable due to the singularity of the Hessian matrix. Also, since the likelihood value is not sensitive to initial $\boldsymbol{\theta}$ values while the estimate for $\boldsymbol{\theta}$ is, we can not trust the $\boldsymbol{\theta}$ estimation.

MLE from Gaussian Correlation Function with One Parameter

Next we move on to the Gaussian correlation function with one parameter equation (3.80). In this section, we still trying to get MLE for both B and θ for multivariate kriging model. This time, we include the design 18 variables in the design matrix first then try another one with only intercept included.

Estimates for B are given in Table 3.20. The estimated Ω is given in equation (3.83). The corresponding estimated θ is 0.6538. For this univariate case, we do not have the singularity problem with the Hessian matrix. The estimation for θ is also pretty stable.

Comparing Table 3.19 and Table 3.20, we can see the estimates for B are very close for multiple and one parameter versions for the Gaussian correlation function. We trust this result more.

Table 3.20. $\hat{\beta}$ for kriging model using equation (3.80)

$\hat{B}$	1	2	3	4	5
1	**1.4483**	**0.9821**	**1.1112**	**0.9899**	**0.5583**
2	0.0008	0.0004	-0.0033	0.0185	-0.0042
3	0.0035	0.0041	0.0094	-0.0009	-0.0151
4	0.0197	0.0018	-0.0183	0.0144	0.0267
5	-0.0012	-0.0036	-0.0141	-0.0184	-0.0013
6	0.0071	0.0073	-0.0311	0.0122	0.0011
7	0.0001	-0.0036	-0.0183	0.0105	-0.0187
8	-0.0132	0.0042	-0.0084	-0.0044	0.0017
9	-0.0079	0.0086	-0.0062	0.0291	-0.0142
10	-0.0140	0.0066	0.0115	-0.0073	-0.0104
11	0.0078	0.0005	0.0018	-0.0029	0.0142
12	-0.0111	0.0111	-0.0173	0.0202	0.0232
13	-0.0018	-0.0007	-0.0072	-0.0048	0.0009
14	0.0127	-0.0106	-0.0046	-0.0086	-0.0100
15	-0.0100	0.0010	0.0120	-0.0096	-0.0097
16	0.0050	-0.0041	0.0005	-0.0056	-0.0070
17	0.0083	-0.0005	0.0011	-0.0260	0.0114
18	-0.0073	0.0071	0.0051	-0.0369	0.0399
19	-0.0012	-0.0027	-0.0072	0.0261	-0.0628

$$\hat{\Psi} = \begin{pmatrix} 0.0004 & -0.0002 & -0.0001 & -0.0000 & -0.0001 \\ -0.0002 & 0.0003 & 0.0000 & -0.0001 & 0.0001 \\ -0.0001 & 0.0000 & 0.0006 & -0.0006 & -0.0003 \\ -0.0000 & -0.0001 & -0.0006 & 0.0015 & -0.0000 \\ -0.0001 & 0.0001 & -0.0003 & -0.0000 & 0.0009 \end{pmatrix} \tag{3.83}$$

Next, we tried the kriging model with design matrix only include the intercept term. Table 3.21 gives the estimated B, and (3.84) gives the estimated Ω. The estimated θ is 0.1653.

Table 3.21. $\hat{B}$ for kriging model with intercept only using (3.80)

$\hat{B}$	1	2	3	4	5
1	1.4483	0.9821	1.1112	0.9899	0.5582

$$\hat{\Psi} = \begin{pmatrix} 0.0013 & -0.0005 & -0.0006 & 0.0004 & 0.0001 \\ -0.0005 & 0.0006 & -0.0002 & 0.0001 & 0.0004 \\ -0.0006 & -0.0002 & 0.0031 & -0.0027 & 0.0005 \\ 0.0004 & 0.0001 & -0.0027 & 0.0054 & -0.0022 \\ 0.0001 & 0.0004 & 0.0005 & -0.0022 & 0.0056 \end{pmatrix} \tag{3.84}$$

Although we got good MLE results using the single variable Gaussian correlation function for kriging model, we find another aspect that is not good. Figure 3.1 shows the likelihood function versus θ. We can see that the likelihood function is strictly increasing along with θ and when θ is large, the increasing is extremely slow. This will create a problem in the MLE estimation since there is no mode in this picture.

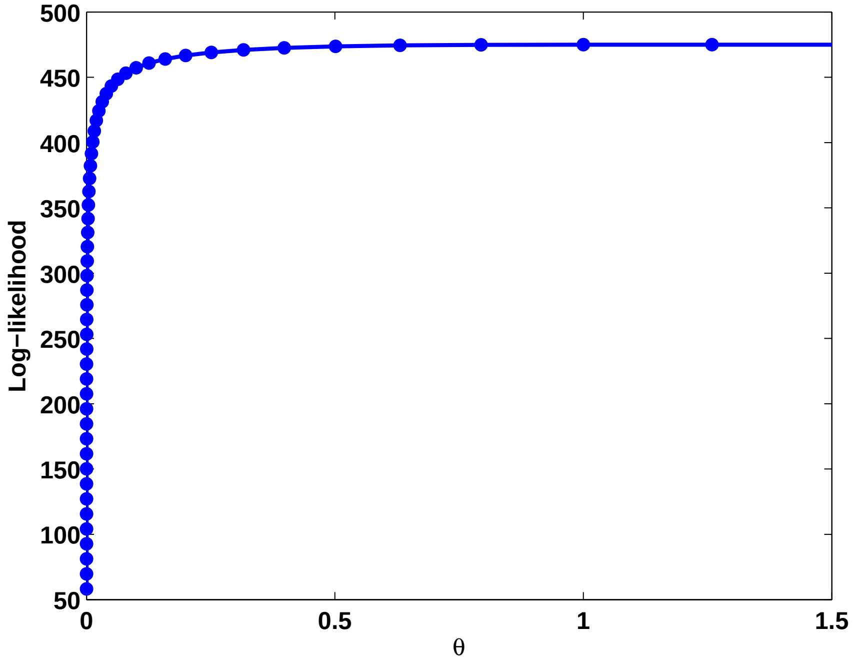


Figure 3.1. *Likelihood v.s. θ.*

Parameter Estimation for Single Variable Gaussian Correlation Function Using Cross-validation

In this section, we will use cross-validation to estimate the unknown parameter θ in the Gaussian correlation function (3.80). In previous results, we can see that the estimated intercept B remains constant for different forms of Gaussian correlation functions and for different design matrix. Hence in what follows, the design matrix will contain only the intercept term.

We searched θ from 0 to 10 on a given grid set. For each θ value, we obtain a leave-one-out cross-validation score for the multivariate kriging model which is discussed in the previous section. Figure 3.2 gives the curve for the CV score versus θ. The discrete CV score achieves the minimum value at $\theta = 0.1585$ which is pretty close to the MLE for the intercept only case. The corresponding estimated B and Ω

are given in Table 3.22 and 3.85.

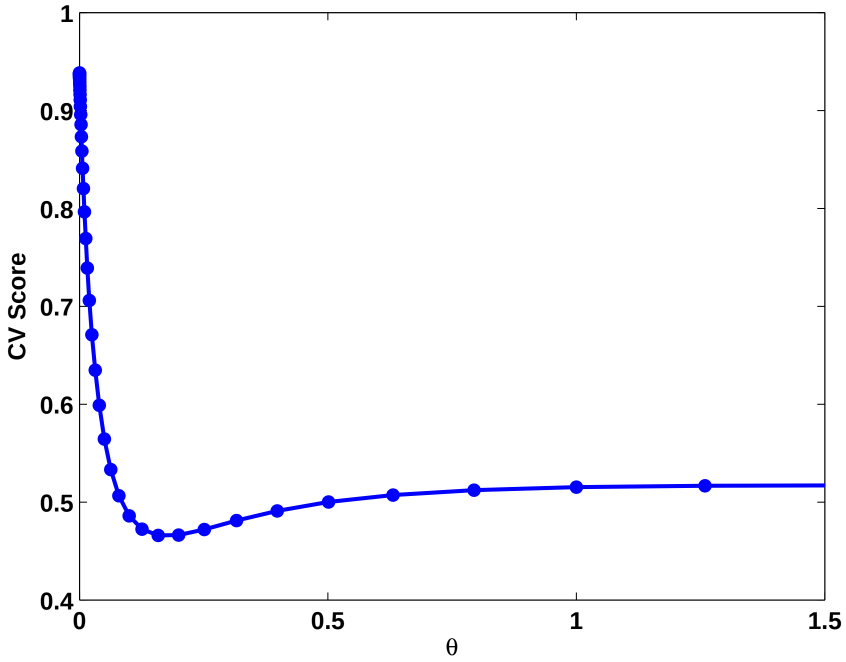


Figure 3.2. *CV Score v.s. θ.*

Table 3.22. $\hat{B}$ for kriging model with intercept only using (3.80) by cross-validation

$\hat{B}$	1	2	3	4	5
1	1.4450	0.9826	1.1138	0.9781	0.5682

$$\hat{\Psi} = \begin{pmatrix} 0.0025 & -0.0011 & -0.0008 & 0.0001 & 0.0001 \\ -0.0011 & 0.0015 & -0.0002 & 0.0000 & 0.0010 \\ -0.0008 & -0.0002 & 0.0050 & -0.0044 & -0.0003 \\ 0.0001 & 0.0000 & -0.0044 & 0.0103 & -0.0033 \\ 0.0001 & 0.0010 & -0.0003 & -0.0033 & 0.0103 \end{pmatrix} \tag{3.85}$$

3.7.3 Functional ANOVA

To investigate the effect of each variable, we use multivariate functional ANOVA which was discussed before to decompose the Gaussian kriging model. From the previous section, the final multivariate kriging model for this case can be written as

$$\hat{\mathbf{Y}}(\mathbf{x}) = \hat{B}_0 + \hat{\mathbf{r}}^T(\mathbf{x})\hat{\mathbf{R}}_r^{-1}(\mathbf{Y} - \mathbf{X}\hat{B}_0) \tag{3.86}$$

where $\hat{B}_0 = (1.4450, 0.9826, 1.1138, 0.9781, 0.5682)$ is a 1 by s vector,

$$\hat{\mathbf{r}}(\mathbf{x}) = (exp(-\hat{\theta}\sum_{i=1}^{d}(x_i - X_{1,i})^2), \cdots, exp(-\hat{\theta}\sum_{i=1}^{d}(x_i - X_{n,i})^2))^T = (r_1(\mathbf{x}), \cdots, r_n(\mathbf{x}))$$

is the Gaussian correlation vector between prediction point **x** and all the design point $X_{j,i}$ where $i = 1, \cdots, d$ and $j = 1, \cdots, n$. $\hat{\mathbf{R}}_r$ is the estimated Gaussian correlation matrix between all the design point which is a n by n matrix. **Y** is the n by s response matrix, $\mathbf{X} = (1, \cdots, 1)$ is a 1 by n matrix. Here, $n = 30$, $d = 18$ and $s = 5$.

The calculation shows that the overall behavior is

$$\int \mathbf{g}\mathbf{g}^T d\mathbf{x} = 5.5787, \quad \mathbf{g}_0\mathbf{g}_0^T = 5.5736,$$

and the total variance is

$$D = \int \mathbf{g}\mathbf{g}^T d\mathbf{x} - \mathbf{g}_0\mathbf{g}_0^T = 0.0051.$$

The total variance for the main effects are

$$\sum_{i=1}^{s} \int \mathbf{g}_i\mathbf{g}_i^T d\mathbf{x} = 0.0049.$$

Hence the overall variance explained by the main/linear effects are: $0.0049/0.0051 = 96\%$. The detailed componentwise information is given in Table 3.23.

From Table 3.23, we can see that the largest componentwise variation comes from the 5th component, which corresponding to the highest RPM (difference) level.

Table 3.23. Componentwise total variation

Source	1	2	3	4	5
${g^{(l)}}^2$	2.0903	0.9653	1.2358	0.9844	0.3029
${g_0^{(l)}}^2$	2.0900	0.9652	1.2348	0.9828	0.3008
$D^{(l)}$	0.0003	0.0001	0.0010	0.0016	0.0020

The next largest variation comes from the 4th component, which corresponding to the second highest RPM level. The third largest variation comes from the third component, which corresponding to the third highest RPM level. The rest component are negligible. We need to control the predictors under the highest, second highest and third highest RPM level.

Table 3.24 shows the component-wise individual main effect total variance. From the above discussion, we need to concentrate on the 5th, the 4th, and the third component. For the 5th component, we noticed that $g_{18}(x_{18})$ has the largest variation. Next in order are $g_{17}(x_{17})$, $g_8(x_8)$, $g_6(x_6)$, and $g_3(x_3)$. For the 4th component, $g_{17}(x_{17})$ has the largest variation. Next in order are $g_{18}(x_{18})$, $g_8(x_8)$, $g_3(x_3)$, and $g_{11}(x_{11})$. For the third component, $g_5(x_5)$ has the largest variation. Next in order are $g_3(x_3)$, $g_6(x_6)$, $g_4(x_4)$, and $g_{11}(x_{11})$. Table 3.25 gives the corresponding componentwise main effect Sobol indices. Table 3.26 gives the overall variation decomposition for each main effect and their corresponding Sobol indices. This variation is just the simple summation across the componentwise direction in Table 3.24. The Sobol indices shows that $\mathbf{g}_{18}(x_{18})$ has the largest variation. Then is $\mathbf{g}_{17}(x_{17})$, then is $\mathbf{g}_3(x_3)$.

Table 3.24. $1000 * \int_0^1 {g_i^{(l)}}^2 dx_i$

	1	2	3	4	5
1	0.0029	0.0000	0.0157	0.0607	0.0396
2	0.0125	0.0026	0.0462	0.0214	0.0045
3	0.0893	0.0031	**0.1444**	**0.1123**	**0.1083**
4	0.0012	0.0003	**0.0828**	0.0006	0.0170
5	0.0126	0.0254	**0.3133**	0.0653	0.0016
6	0.0000	0.0021	**0.1317**	0.0369	**0.1262**
7	0.0056	0.0007	0.0118	0.0226	0.0334
8	0.0000	0.0136	0.0492	**0.2397**	**0.1345**
9	0.0594	0.0114	0.0082	0.0031	0.0410
10	0.0044	0.0001	0.0003	0.0190	0.0138
11	0.0112	0.0139	**0.0699**	**0.1156**	0.0167
12	0.0079	0.0006	0.0000	0.0077	0.0260
13	0.0485	0.0270	0.0015	0.0017	0.0223
14	0.0153	0.0001	0.0404	0.0849	0.0008
15	0.0083	0.0000	0.0024	0.0137	0.0491
16	0.0019	0.0006	0.0014	0.0812	0.0208
17	0.0171	0.0096	0.0398	**0.3630**	**0.3484**
18	0.0024	0.0015	0.0501	**0.2939**	**0.9735**

Table 3.25. Componentwise main effect Sobol indices

	1	2	3	4	5
1	0.0092	0.0003	0.0152	0.0377	0.0195
2	0.0394	0.0216	0.0446	0.0133	0.0022
3	0.2809	0.0252	**0.1394**	**0.0698**	**0.0532**
4	0.0037	0.0022	**0.0799**	0.0004	0.0083
5	0.0397	0.2067	**0.3023**	0.0405	0.0008
6	0.0000	0.0171	**0.1271**	0.0229	**0.0620**
7	0.0175	0.0058	0.0114	0.0140	0.0164
8	0.0000	0.1104	0.0475	**0.1489**	**0.0661**
9	0.1868	0.0927	0.0079	0.0019	0.0201
10	0.0138	0.0010	0.0003	0.0118	0.0068
11	0.0351	0.1129	**0.0675**	**0.0718**	0.0082
12	0.0250	0.0050	0.0000	0.0048	0.0128
13	0.1526	0.2195	0.0014	0.0010	0.0110
14	0.0481	0.0008	0.0389	0.0527	0.0004
15	0.0261	0.0001	0.0023	0.0085	0.0241
16	0.0059	0.0049	0.0014	0.0504	0.0102
17	0.0538	0.0782	0.0384	**0.2255**	**0.1713**
18	0.0076	0.0122	0.0483	**0.1825**	**0.4785**
sum	0.9453	0.9166	0.9738	0.9585	0.9720

Table 3.26. Overall variance decomposition for each main effect and their Sobol indices

	$1000 * \int \mathbf{g}_i \mathbf{g}_i^T d\mathbf{x}$	Overall Sobol's indices$\int \mathbf{g}_i \mathbf{g}_i^T d\mathbf{x}/D$
1	0.1190	0.0221
2	0.0873	0.0162
3	0.4575	**0.0850**
4	0.1019	0.0189
5	0.4181	**0.0777**
6	0.2969	0.0552
7	0.0741	0.0138
8	0.4369	**0.0812**
9	0.1231	0.0229
10	0.0376	0.0070
11	0.2272	0.0422
12	0.0423	0.0079
13	0.1009	0.0188
14	0.1415	0.0263
15	0.0735	0.0137
16	0.1059	0.0197
17	0.7779	**0.1446**
18	1.3214	**0.2456**

Figures 3.3 to 3.8 give the main effect functions $g_i^{(l)}(x_i)$ for $i = 1, \cdots, d$ and each components $l = 1, \cdots, s$.

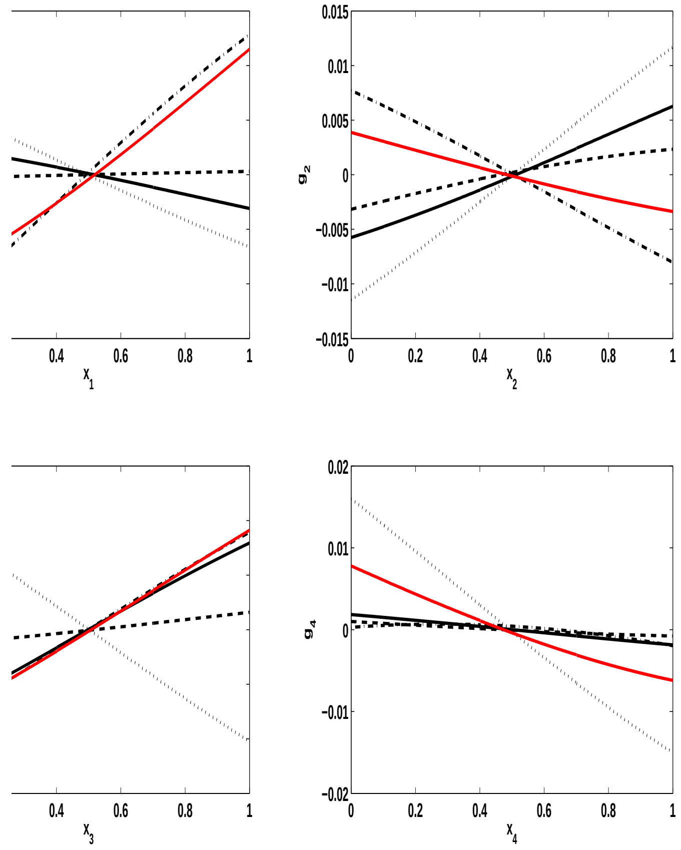


Figure 3.3. *Functional ANOVA decomposition for x_1 to x_4. -: $1st$ component; –: $2nd$ component; . $3rd$ component; -.- $4th$ component; red -: $5th$ component*

3.8 Summary

In this chapter, a multivariate kriging model has been proposed. Using matrix normal theory, we developed an estimation procedure the the multivariate kriging model. For given parameters in variance-covariance matrix, the MLE for coefficient

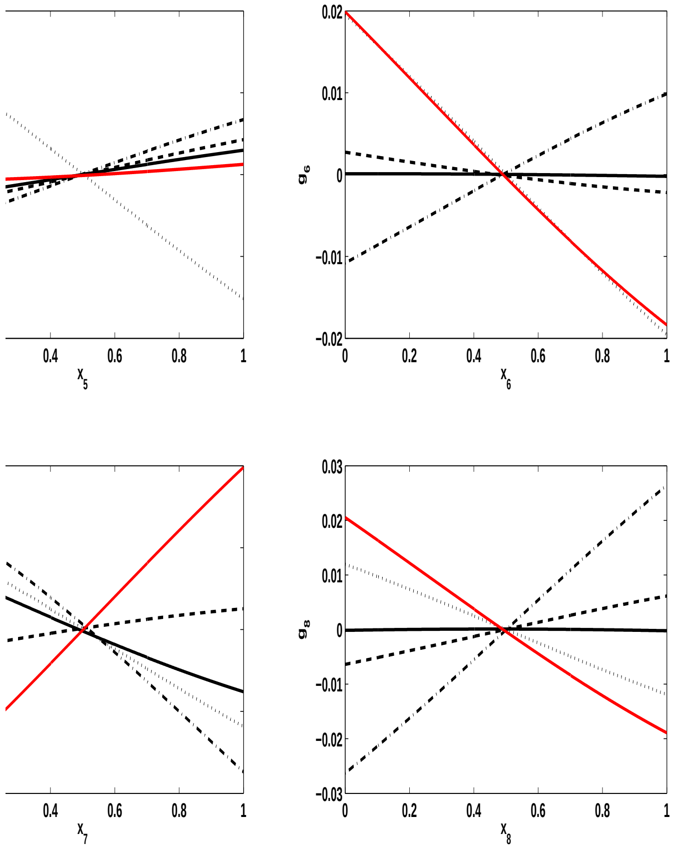


Figure 3.4. *Functional ANOVA decomposition for x_5 to x_8. -: $1st$ component; –: $2nd$ component; . $3rd$ component; -.- $4th$ component; red -: $5th$ component*

has been proved to have a closed form, which is given in (3.37). There is no closed form for the estimates for the coefficients and the parameters in variance-covariance matrix. Hence we proposeed using Newton-Ralphson algorithm to optimize the corresponding likelihood function and obtain the MLE of the parameters.

To interpret the resulting multivariate Gaussian model, we extended the functional ANOVA decomposition into multivariate version. For Gaussian correlation function, we derived closed forms for the multivariate functional ANOVA decompo-

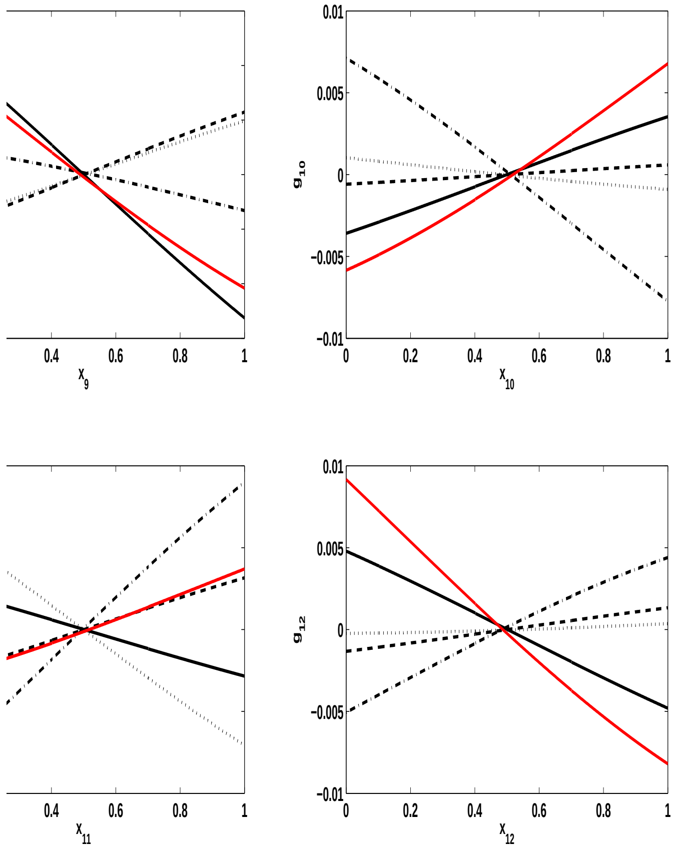


Figure 3.5. *Functional ANOVA decomposition for x_9 to x_{12}. -: 1st component; –: 2nd component; . 3rd component; -.- 4th component; red -: 5th component*

sition.

To validate the proposed modeling procedures, a simulation study has been conducted to similar settings in the engine noise example. The simulation results show that the proposed estimation procedure performs very well even in the very large noise level situation.

To demonstrate the advantage of the multivariate kriging model, we compared it with the kriging model with single response. We use leave-one-out method

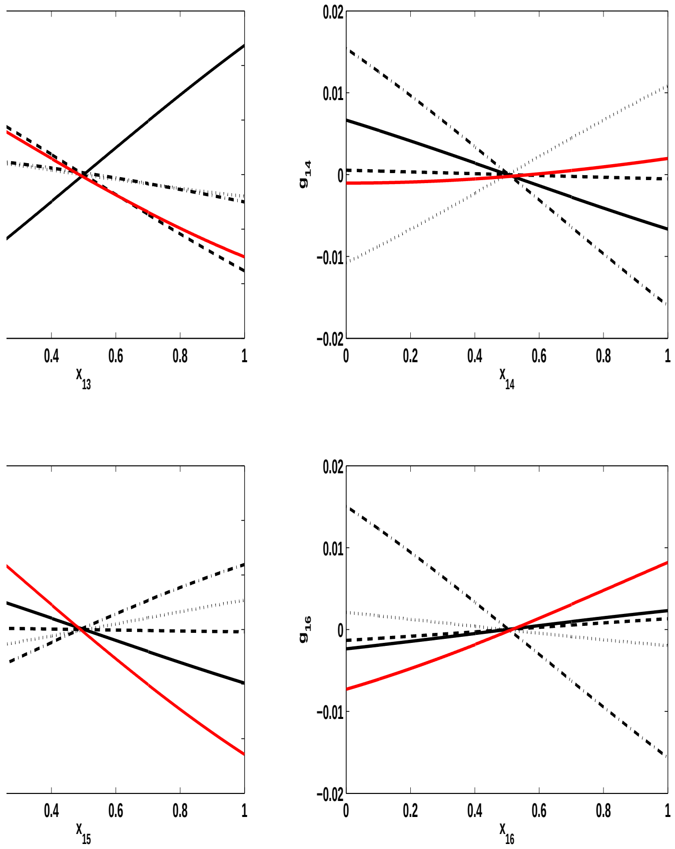


Figure 3.6. *Functional ANOVA decomposition for x_{13} to x_{16}. -: $1st$ component; –: $2nd$ component; . $3rd$ component; -.- $4th$ component; red -: $5th$ component*

to get the prediction error for both models. We performed the simulation for 1000 times and get the average of the prediction error. The result shows that the multivariate kriging model has more prediction power than the kriging model with single response.

We conducted a detailed analysis to the engine noise example using the proposed multivariate kriging model and functional ANOVA decomposition. In this example, there are 18 design factors in total. The response is the difference of log of

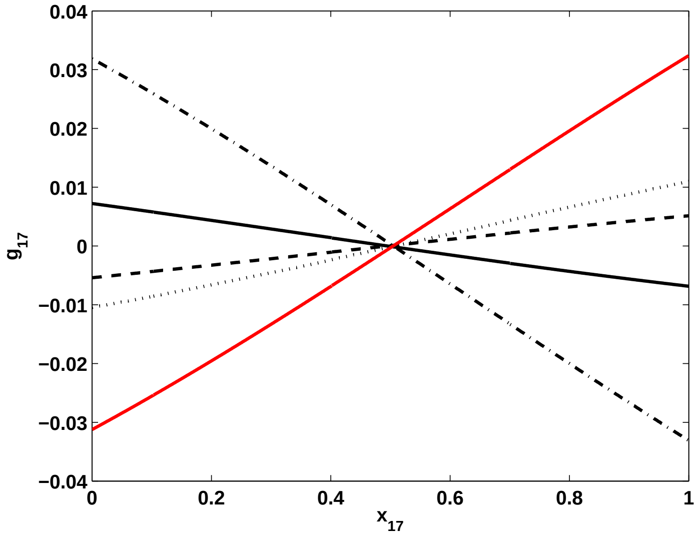


Figure 3.7. *Functional ANOVA decomposition for* x_{17}. *-:* $1st$ *component; –:* $2nd$ *component; .* $3rd$ *component; -.-* $4th$ *component; red -:* $5th$ *component*

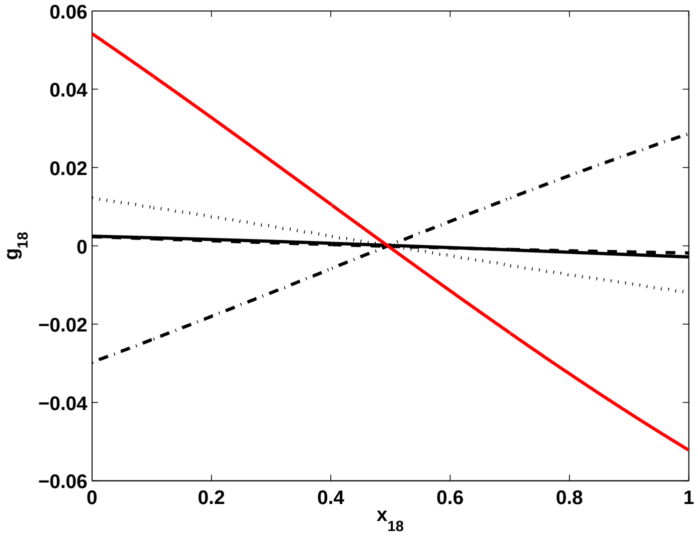


Figure 3.8. *Functional ANOVA decomposition for* x_{18}. *-:* $1st$ *component; –:* $2nd$ *component; .* $3rd$ *component; -.-* $4th$ *component; red -:* $5th$ *component*

engine noise between adjacent RPM levels. We are interested in finding out the most influential factor(s), and understand how each variable affects the responses.

The sensitivity analysis is performed first by regression partial sum of squares. The RPM is included as a predictor in this linear regression. Not surprisingly, this approach found that RPM is the most influential factor among all the predictors. The rough importance rank ordering for the rest of predictors are:

$$x_{18}, x_6, x_3, x_{19}^2, x_{18}x_{19}, x_1x_{19}, x_{13}x_{19}, x_1x_{15}, x_{17}x_{19}, x_{11}x_{19}, x_{11}x_{17}.$$

From the regression coefficients, we need to minimize the design variables x_3, x_1, x_{17}, x_{11}. Also, we need to maximize x_{18}, x_6, x_{13}, and x_1. From the analysis, we know that the influence of RPM is much higher than the rest of the predictors. In order to study the rest of the variables in detail, we need to separate RPM with all other predictors.

We further apply the multivariate functional ANOVA decomposition to this example. The linear effects for all the predictors are very small compare to the intercept term. Hence we further considered a multivariate kriging model that includes the intercept term only. The multivariate functional ANOVA decomposition result tells us that the largest componentwise variation comes from the 5th component, which corresponding to the highest RPM (difference) level. The next largest variation comes from the 4th component, which corresponding to the second highest RPM level. The third largest variation comes from the third component, which corresponding to the third highest RPM level. The rest component are negligible. We need to control the predictors under the highest, second highest and third highest RPM level. Also, for the 5th component, we noticed that x_{18} generates the largest variation. Next in order are x_{17}, x_8, x_6, and x_3. For the 4th component, x_{17} has the largest variation. Next in order are x_{18}, x_8, x_3, and x_{11}. For the third component, x_5 has the largest variation. Next in order are x_3, x_6, x_4, and x_{11}. The result from Sobol indices shows that x_{18} has the largest variation. Then is x_{17}, then is x_3.

The methodology in this chapter can be directly applied to any computer experiments with output data which can be consider as multivariate responses. The example in this chapter is just a typical one of such computer experiments.

Chapter 4

Modeling Computer Experiment with Functional Response by FLM

4.1 Introduction

In this chapter, we will consider the computer experiment with intensive functional response. As discussed before, functional data is the data which are collected over a certain range with a regular frequency. In Chapter 3, we studied the functional response with sparse sampling rate. But usually, functional data is fairly dense compared to longitudinal data and are collected at even space. These kind of data are usually either collected by computer, or are the output of some computer code which are run with various input variable at a set of even grid points.

As advances in modern technology and device of collecting data, people can easily collect and store functional data with intensive sampling rate. Functional data analysis is becoming popular in various research fields. Many techniques for the analysis of data with functional nature which allows the variability exploration in samples of curves and let the research of common structures among the curves or groups of them have been introduced. The **Valvetrain** example (Example 1.3) which is introduced in chapter 1 is a typical real case of the intensive functional response. In this example, the valve motion error is simulated over the crank angle degree range $(0, 720)$ at almost each degree. For each case in the 16 design situations in Table 1.5, we have 657 data points along the functional direction. Hence we can see clearly a curve for each design case. Figure 4.1 shows the computer simulated motion errors of the valve compared to the prescribed motion by the camshaft for all the 16 designs listed in Table 1.5. We can see from these plots that the errors within range (0,90) and (450,720) are very small compared to the errors within the range (90,450) for all the design cases.

Table 4.1 shows the final design matrix $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_9)$, where $\mathbf{X}_i = (x_{1,i}, x_{2,i}, \cdots, x_{16,i})^T$. Note that in Table 1.5, the last column has three levels. In order to make it similar with other columns in the matrix, we split it into two columns, one is the indicator of Ture/False if it equal to 1, the other is the indicator of Ture/False if it equal to 2. X_1 is the intercept term. The eight predictors are:

X_2: H.S., Cylinder Head stiffness;

X_3: R.S., Rocker Arm stiffness,;

X_4: L.A., Hydraulic Lash Adjuster;

X_5: C.P., Cam Phasing, Clearance;

X_6: C.L., Clearance;

X_7: S.H., Spring;

X_8: R.H.1, Ramp Height 1.

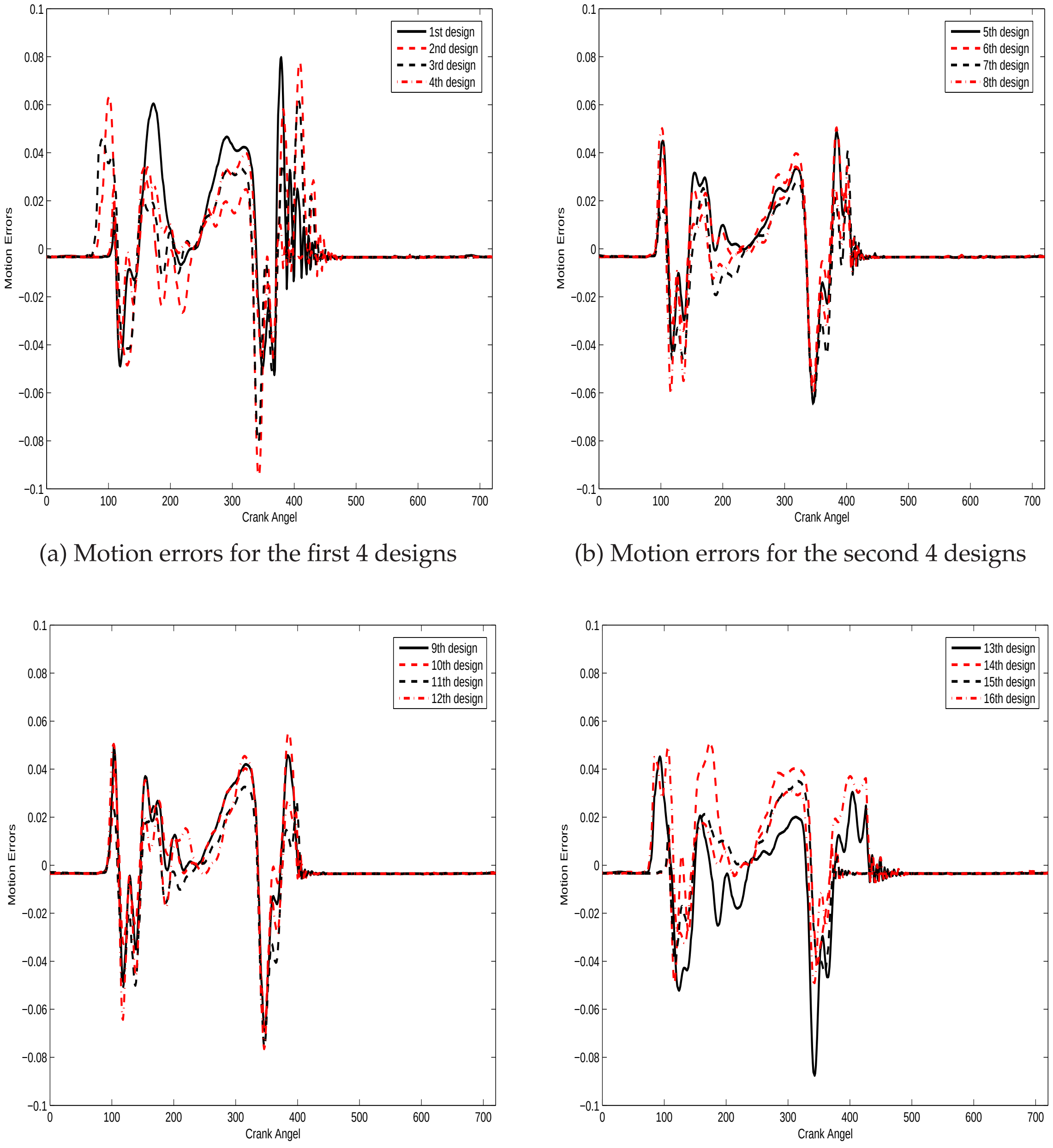
X_9: R.H.2, Ramp Height 2.

Intuitively, we can get the ordinary least square estimates

$$\hat{\boldsymbol{\beta}}_{OLS}(u_j) = (\mathbf{X'X})^{-1}\mathbf{X'Y}(u_j)$$

at each crank angle point u_j, $j = 1, \cdots, J$ for the coefficients $\boldsymbol{\beta}$ in the following model:

$$y_i(u_j) = \mu(\mathbf{x}_i; u_j) + \varepsilon_{ij} = \boldsymbol{\beta}(u_j)^T\mathbf{x}_i + \varepsilon_{ij}, \tag{4.1}$$



(a) Motion errors for the first 4 designs

(b) Motion errors for the second 4 designs

(c) Motion errors for the third 4 designs

(d) Motion errors for the last 4 designs

Figure 4.1. *Valve motion errors for 16 valvetrain designs listed in Table 1.5*

Table 4.1. *Final design matrix* **X** *for Valvetrain*

$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{X}_5$	$\mathbf{X}_6$	$\mathbf{X}_7$	$\mathbf{X}_8$	$\mathbf{X}_9$
1	0	0	0	0	1	1	0	0
1	1	0	0	0	1	0	0	1
1	0	1	0	0	0	1	0	1
1	1	1	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0
1	1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1	0
1	1	1	1	0	1	1	1	0
1	0	0	0	1	0	0	1	0
1	1	0	0	1	0	1	1	0
1	0	1	0	1	1	0	1	0
1	1	1	0	1	1	1	1	0
1	0	0	1	1	1	1	0	1
1	1	0	1	1	1	0	0	0
1	0	1	1	1	0	1	0	0
1	1	1	1	1	0	0	0	1

where $\mathbf{Y}(u_j) = (y_1(u_j), \cdots, y_n(u_j))^T$, $i = 1, \cdots, n = 16$, $\mathbf{x}_i = (x_{i,1}, x_{i,2}, x_{i,p})$, $p = 9$, and

$$\boldsymbol{\beta}(u_j) = (\beta_0(u_j), \beta_1(u_j), \cdots, \beta_p(u_j))^T.$$

In model (4.1), we model the overall mean function as:

$$\mu(\mathbf{x}; u) = \boldsymbol{\beta}(u)^T \mathbf{x}. \tag{4.2}$$

Figure 4.2 shows the OLS estimates for the functional coefficients within crank angle $(0, 720)$. Again, the amplitude of these functional coefficients are very small within ranges $(0, 90)$ and $(450, 720)$. Furthermore, we can see that all the 9 coefficient components amplitude are about on the same level within the crank angle $(360, 450)$. Meanwhile, in the crank angle $(90, 360)$, there are several coefficient components with variation larger than the other components. We can apply the functional linear model to estimate the mean function for the valve motion error within crank angle $(360, 450)$. To get a parsimonious model for the valve motion error within crank angle $(90, 360)$, we will use the partial functional linear model to estimation the mean function in the next chapter. After we get the estimation for the mean function, we can get the residual at each observation point. We then will develop a spatial-temporal model to get a smooth surface which will interpolate each residual point. Adding this surface to the smooth mean function, we then have a smooth surface which will passes through each observation point.

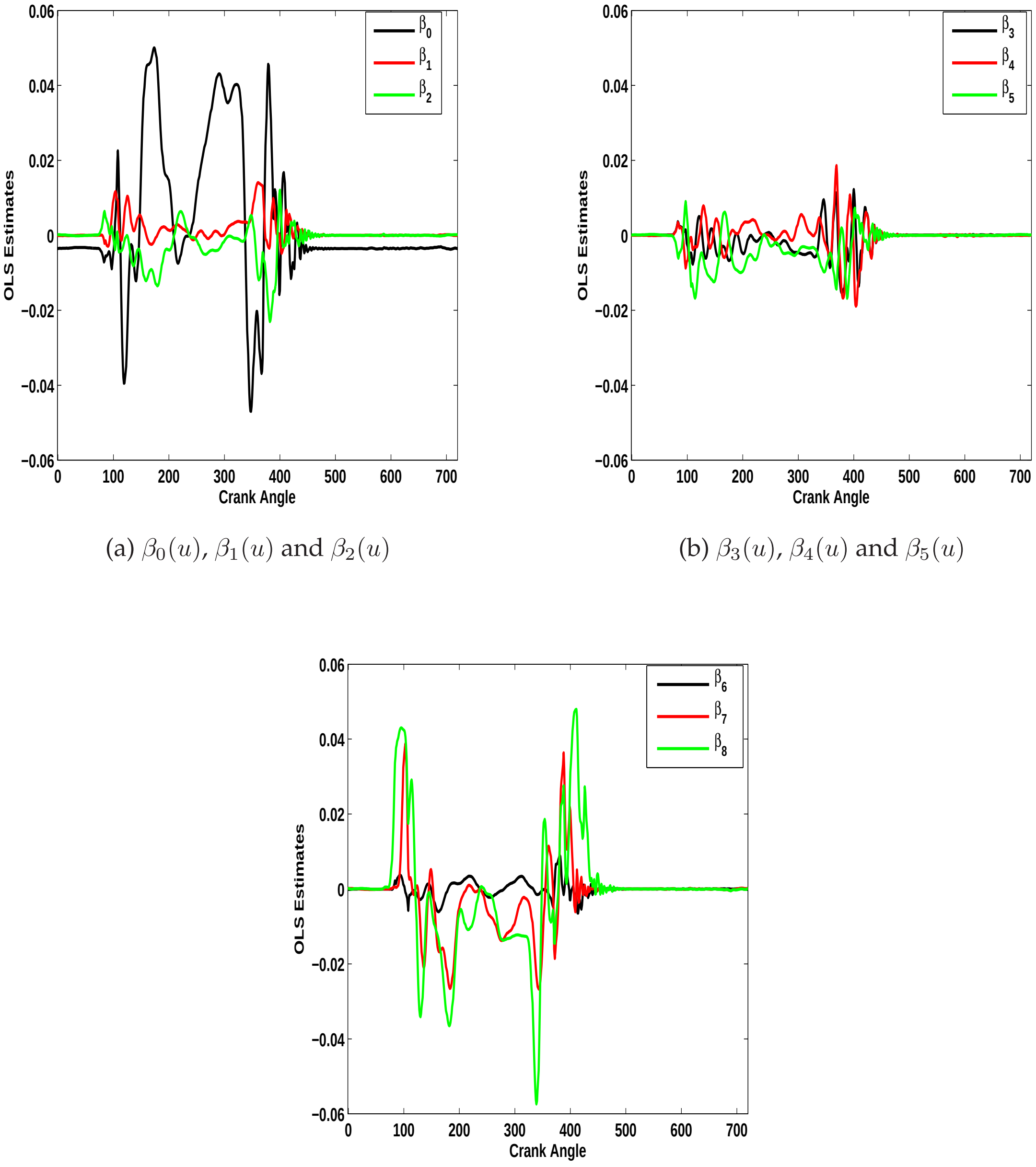
4.2 Estimating Procedure for Functional Linear Model

Let n denote the number of observations for $(\mathbf{x}_i, y_i)$ and let J denote the number of time points for each observation. Here, we assume the number and location of time point are the same for each observation. This is often the case in the computer experiments since all the data are collected by computer or the output of the computer code.

Consider the ith observed data pair at the jth time index (or at the time u_j) $(\mathbf{x}_i, y_i(u_j))$, $i = 1, \cdots, n$ where $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,p})^T$ is the predictor set at ith observed data pair. Here, we assume the predictor set do not change with time. Then the functional linear model is:

$$\begin{aligned} y_i(u_j) &= \mu(\mathbf{x}_i, u_j) + \varepsilon_{ij} \\ &= \mathbf{X}_i \boldsymbol{\beta}(u_j) + \varepsilon_{ij} \\ &= \beta_0(u_j) + \boldsymbol{\beta}_1(u_j)^T \mathbf{x}_i + \varepsilon_{ij} \end{aligned} \tag{4.3}$$

where $\mathbf{X}_i = (1, \mathbf{x}_i^T)$, $i = 1, \cdots, n$ and $j = 1, \cdots, J$. For a given time index j, the



(a) $\beta_0(u)$, $\beta_1(u)$ and $\beta_2(u)$

(b) $\beta_3(u)$, $\beta_4(u)$ and $\beta_5(u)$

(c) $\beta_6(u)$, $\beta_7(u)$ and $\beta_8(u)$

Figure 4.2. *OLS estimates for the functional coefficients within crank angle region (0,720)*

functional coefficient vector is:

$$\boldsymbol{\beta}_1(u_j) = (\beta_1(u_j), \beta_2(u_j), \cdots, \beta_p(u_j))^T$$

Note that both the coefficients $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_1^T)^T$ and the predictor **x** are the functions of another index variable u (for example, time). But the random error ε has mean zero.

Our purpose is to get the estimations for the coefficient functions $\boldsymbol{\beta}(u)$. We use two step estimation procedure–OLS estimates at each time point and local linear smoothing along time direction for each component of functional coefficients.

4.2.1 OLS estimates at each time point

For any given j (u_j), ordinary least square estimators can be used to estimate the initial point coefficients across all the observations at jth time index:

$$\tilde{\boldsymbol{\beta}}(u_j) = (\tilde{\beta}_0(u_j), \tilde{\boldsymbol{\beta}}_1(u_j)^T)^T \tag{4.4}$$

which is:

$$\tilde{\boldsymbol{\beta}}(u_j) = \mathbf{A}\mathbf{Y}(u_j) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}(u_j) \tag{4.5}$$

where

$$\mathbf{Y}(u_j) = (y_1(u_j), y_2(u_j), \cdots, y_n(u_j))^T$$

and

$$\mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix}$$

is the corresponding design matrix with the first subscript as the index of the observations and second subscript as the index of the predictors. And:

$$\mathbf{A} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{4.6}$$

The estimated value for $Y(u_j)$ is:

$$\hat{\mathbf{Y}}(u_j) = P_{\mathbf{Y}_j}\mathbf{Y}(u_j) = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}(u_j), \tag{4.7}$$

where

$$P_{Y_j} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{4.8}$$

is the projection (hat) matrix for $\mathbf{Y}(u_j)$.

4.2.2 Local linear smoothing for each coefficient component

After we get all the initial point estimations for the $\boldsymbol{\beta}(u_j)$ from the OLS estimator (4.5) for $j = 1, \cdots, J$, the next step is to do smoothing for each $\tilde{\beta}_j$ along time direction. The smoothing step is necessary due to the following reasons. First of all, this smoothing step can give smooth estimates for the underlying smooth coefficient functions. Secondly, since the first step estimator do not use the information from the neighboring time point, hence they are inefficient. Smoothing each of the first step estimator along time direction enables the estimator to pool information from the neighboring time points hence it will improve the efficiency of the raw estimates. Thirdly, we may also want to estimate the values of the coefficient curves at non-design points. And the smoothing is one-dimensional for each individual coefficient function hence for different components of coefficient functions, different amount of smoothing can be conducted and any existing smoothing technique can be utilized. At here, the local linear regression with Epanechnikov kernel will be used to handle the smoothing step.

Consider the local linear estimation for smooth coefficient function $\beta_j(u)$ where $j = 0, 1, \cdots, p$. Since $\beta_j(u)$ is smooth, applying Taylor expansion for $\beta_j(u)$ in a neighborhood of u (for example, for u_i close to u):

$$\beta_j(u_i) \approx \beta_j(u) + (u_i - u)\beta_j'(u) = \mathbf{u}_i^T\mathbf{B}_j(u) \tag{4.9}$$

where $\mathbf{u}_i = (1, (u_i - u))^T$ and $\mathbf{B}_j(u) = (B_{j,1}(u), B_{j,2}(u))^T = (\beta_j(u), \beta'_j(u))^T$. Our purpose is to get the estimation for all the functions $\mathbf{B}_j(u)$ where $j = 0, 1, \cdots, p$. Here, more specifically, we need an estimation for the first component function $\beta_j(u) = B_{j,1}(u)$. Intuitively, datum points close to u will have more influence about $\beta_j(u)$. This suggests using a locally weighted polynomial regression:

$$\min_{\mathbf{B}_j(u)} \sum_{i=1}^{J} (\tilde{\beta}_j(u_i) - \mathbf{u}_i \mathbf{B}_j(u))^2 K_h(u_i - u), \tag{4.10}$$

where $K_h(x) = h^{-1}K(x)$ and $K(x)$ is the kernel function. Denote $W = diag(K_h(u_1 - u), \cdots, K_h(u_J - u))$, and $\mathbf{U}$ as the design matrix associated with $\mathbf{u}_i, i = 1, J$, then the solution to locally weighted least squares (4.10) is:

$$\hat{\mathbf{B}}_j(u) = (\mathbf{U}'W\mathbf{U})^{-1}\mathbf{U}'W\Gamma_j = P_{\Gamma_j}\Gamma_j \tag{4.11}$$

where $\Gamma_j = (\tilde{\beta}_j(u_1), \cdots, \tilde{\beta}_j(u_J))^T$. The estimator for the regression function $\beta_j(u)$ is:

$$\hat{\beta}_j(u) = \hat{\mathbf{B}}_{j,1}(u)$$

Furthermore, an estimator for the 1st order derivative of $\beta_j(u)$ at u is:

$$\hat{\beta}'_j(u) = \hat{\mathbf{B}}_{j,2}(u)$$

The choice of kernel function is not sensitive to the estimation of $\beta_j(u)$. Since the symmetric kernel function has many advantages over asymmetric functions, the most significant one is it will have better asymptotic properties, hence people prefer the symmetric probability density function. The most commonly used kernel function is the Gaussian kernel function:

$$K(t) = \frac{1}{\sqrt{2\pi}} exp(-t^2/2). \tag{4.12}$$

In Fan (1993), he showed that the best kernel function should be the Epanechnikov kernel:

$$K(t) = 0.75(1 - t^2)_+, \tag{4.13}$$

where $t_+ = t$ when $t \geq 0$ and $t_+ = 0$ when $t < 0$. Here, we use Epanechnikov kernel.

The smoothing bandwidth h is crucial in local linear smoothing. Ruppert et al. (1995) showed that for the local degree p, the optimal bandwidth which minimizes the mean integrate sum error (MISE):

$$MISE = E[\int (\hat{m}(x;h) - m(x))^2 f(x)dx]$$

is:

$$h_{MISE} \approx [\frac{(p+1)(p!)^2 R(K_p) \int v(x)dx}{2\mu_{p+1}(K_p)^2 \int m^{(p+1)}(x)^2 f(x)dxn}]^{1/(2p+3)}, \tag{4.14}$$

where $\mu_l(K) = \int u^l K(u)du$, $R(K) = \int K(u)^2 du$, $v(x)$ is the variance, $m(x)$ is the mean smooth function we want to get, $f(x)$ is the density for X. They use plug-in method to find the estimate for the integrals in above expression. After some calculation, they found that the asymptotic optimal bandwidth for local linear least squares estimate is:

$$h_{AMISE} \propto C_1(K)$$

where $C_1 = 15^{1/5}$ for Epanechnikov kernel. In this research, we use the optimal bandwidth selection subroutine which is provided by Ruppert to calculate the optimal bandwidth for each functional coefficient component β_j. This allows us to use different smoothing parameters for different components of coefficients. Note that this subroutine provides the asymptotic optimal bandwidth for Gaussian kernel for a given dataset. We need to convert this optimal bandwidth to the bandwidth for the Epanechnikov kernel. From Table 4.2 (Ruppert et al. (1995)), we know that $C_1 = (1/(2\sqrt{\pi}))^{1/5}$ for Gaussian kernel. The relationship for the optimal bandwidth between Gaussian kernel (h_2) and Epanechnikov kernel (h_1) is given by:

$$h_1 = (30\sqrt{\pi})^{1/5} h_2 \tag{4.15}$$

Table 4.2. *Kernel-Dependent Constants*

Kernel	Epanechnikov	Biweight	Normal
$C_1(K)$	$15^{1/5}$	$35^{1/5}$	$\{1/(2\sqrt{\pi})\}^{1/5}$

4.2.3 Effect of the Smoothing Step

The effect of the second step in the functional linear model is tosmooth the ordinary least square estimator of the functional coefficients in the model in the time direction. From the Mean Square Error (MSE) of the fitted coefficient point of view, the simulation result shows the necessaries of this smoothing step.

Define the following MSE:

$$\begin{aligned} MSE_i^{(1)} &= \frac{1}{n_i}\sum_{j=1}^{n_i}(\beta_i(u_j) - \tilde{\beta}_i(u_j))^2 \\ MSE_i^{(2)} &= \frac{1}{n_i}\sum_{j=1}^{n_i}(\beta_i(u_j) - \hat{\beta}_i(u_j))^2 \end{aligned} \tag{4.16}$$

where the superscript denotes the step number ("1" refers to OLS estimates, "2" refers to the smoothed fit) and the subscript stands for the different parameters, $i = 0, \cdots, p$ and n_i is the number of points in u direction. From the experiment for the functional linear model in this chapter, we can see that $MSE^{(2)}$ is significantly less than $MSE^{(1)}$, which shows the importance for the smoothing step in the sense of reducing the MSE of the estimates.

But notice that the smoothing procedure is done on the direction of time for each functional coefficient, which is not on the direction of the original observed response, so we expect a certain degree of distortion for the fitted original response. We need to balance the prediction power and the goodness of fit.

4.3 Simulation for local linear regression

In this simulation study, we generated a random sample of size $n = 500$ from the following model

$$Y = 4\sin(2\pi X) + \epsilon,$$

where $X \sim U(0,1)$, the uniform distribution over $[0,1]$, and $\epsilon \sim N(0,1)$. We use local least squares approach with Epanechnikov kernel to estimate the regression function $m(x) = E(Y|X = x)$.

4.3.1 Effective number of parameters

For a random sample $(X_1, Y_1), \cdots, (X_n, Y_n)$ i.i.d. with conditional mean function as $m(x) = E(Y|X = x)$, regression between X and Y, what we want to estimate is the mean function $m(x)$ where:

$$E(Y|X = x) = m(x)$$

where we assume the mean function $m(x)$ is a smooth curve.

Using Taylor expansion around x, we have the following local linear approximation:

$$m(X_i) \approx m(x) + m'(x)(X_i - x) = \beta_0(x) + \beta_1(x)(X_i - x) = \mathbf{x}_i^T \boldsymbol{\beta}(x) \tag{4.17}$$

where $\mathbf{x}_i = (1, X_i - x)^T$, $m(x) = \beta_0(x)$, $m'(x) = \beta_1(x)$, and $\boldsymbol{\beta}(x) = (\beta_0(x), \beta_1(x))^T$.

So $\beta_0(x)$ is an estimator for the mean function $m(x)$. Using kernel function as the weight, we attempt to minimize the following square loss function at each x point:

$$\sum_{i=1}^{n} (Y_i - \mathbf{x}_i^T \boldsymbol{\beta}(x))^2 K_h(X_i - x),$$

where K is the kernel function. In this part, we choose Epanechnikov kernel.

The minimizer for the above summation is the least squares estimate(LSE) for $\boldsymbol{\beta}(x)$:

$$\hat{\boldsymbol{\beta}}(x) = (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}^T W \mathbf{Y} \tag{4.18}$$

where $W = diag(K_h(x - X_i))$, $\mathbf{X} = (\mathbf{x}_1^T, \cdots, \mathbf{x}_n^T)^T$, and $\mathbf{Y} = (Y_1, \cdots, Y_n)^T$.

The predicted value at X_i is:

$$\hat{Y}_i = \hat{m}(X_i) = \hat{\beta}_0(X_i) = (1, 0)(\mathbf{X}_0^T W_0 \mathbf{X}_0)^{-1} \mathbf{X}_0^T W_0 \mathbf{Y} = S_i^T(h)\mathbf{Y}$$

where $\mathbf{X}_0 = (\mathbf{x}_1^0, \cdots, \mathbf{x}_n^0)^T$, $\mathbf{x}_j^0 = (1, X_j - X_i)$, $W_0 = diag(K_h(X_1 - X_i), \cdots, K_h(X_n - X_i))$, and $S_i^T(h) = (1, 0)(\mathbf{X}_0^T W_0 \mathbf{X}_0)^{-1} \mathbf{X}_0^T W_0$ is a one by n row vector. Finally, the smoothing matrix is:

$$S(h) = (S_1, \cdots, S_n)^T,$$

where $\hat{\mathbf{Y}} = S(h)\mathbf{Y}$. Hence, the effective number of parameters is:

$$df(h) = trace(S(h)).$$

We choose the following bandwidth:

$$h = 0.05 * 1.1^j,$$

where $j = 0, 1, \cdots, 30$. The corresponding effective number of parameters versus bandwidth is given in graph 4.3. We can see with increasing of bandwidth, the effective number of parameters will decrease. This probably will result in the increasing of mean square error.

4.3.2 Bandwidth selection

We use Ruppert, Sheather and Wand (1995, JASA) plug-in method to select an optimal bandwidth. Using equation(4.18), we can get the local linear least squares

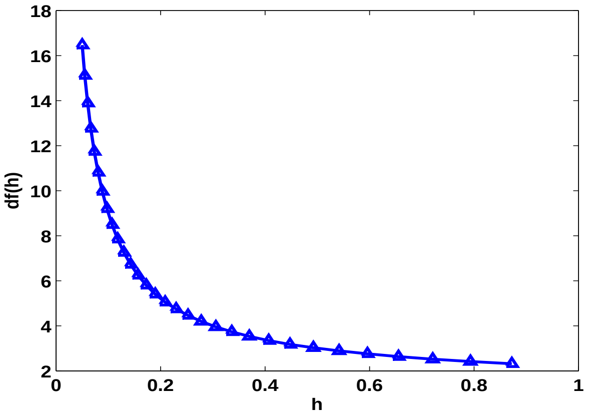


Figure 4.3. *Effective number of parameters*

estimate for $m(x)$ is $\hat{\beta}_0(x)$. To get the estimation variance, consider the variance-covariance matrix for $\hat{\boldsymbol{\beta}}(x)$:

$$cov(\hat{\boldsymbol{\beta}}(x)) = (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}^T W \Sigma W \mathbf{X} (\mathbf{X}^T W \mathbf{X})^{-1},$$

where $\Sigma = cov(\mathbf{Y})$. Since in this simulation, $Y_1, \cdots, Y_n$ are independent with constant variance, $\Sigma = \sigma^2 \mathbf{I}$. The covariance matrix becomes:

$$cov(\hat{\boldsymbol{\beta}}(x)) = \sigma^2 (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}^T W^2 \mathbf{X} (\mathbf{X}^T W \mathbf{X})^{-1}.$$

To estimate σ^2, we use MSE to approximate it. Note that at here,

$$\hat{\sigma}^2 = MSE = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{df} = \frac{\sum_{i=1}^n (Y_i - \hat{\beta}_0(X_i))^2}{df},$$

where $\hat{\beta}_0(X_i)$ can either be calculated at each observation point or can be interpolated from the grid points $(x_i, \hat{\beta}_0(x_i)), i = 1, \cdots, ngrid$. *df* is the effective number of parameters for the corresponding bandwidth. Due to the asymptotic normal property of the local least squares estimate, the 95% confidence interval of the resulting estimate is:

$$\hat{\beta}_0(x) \pm 2SD(\hat{\beta}_0(x)).$$

Using the subroutine provided by Steve Marron (http://www.stat.unc.edu/postscript/papers and the relationship (4.15), the optimal bandwidth for Epanechnikov kernel for the generated dataset is:

$$h_1 = 0.079,$$

and figure 4.4 shows the regression function together with 95% confidence interval. From this plot, we can see that the result is good. Near the boundary, the confidence interval is gradually adjusted to broader range.

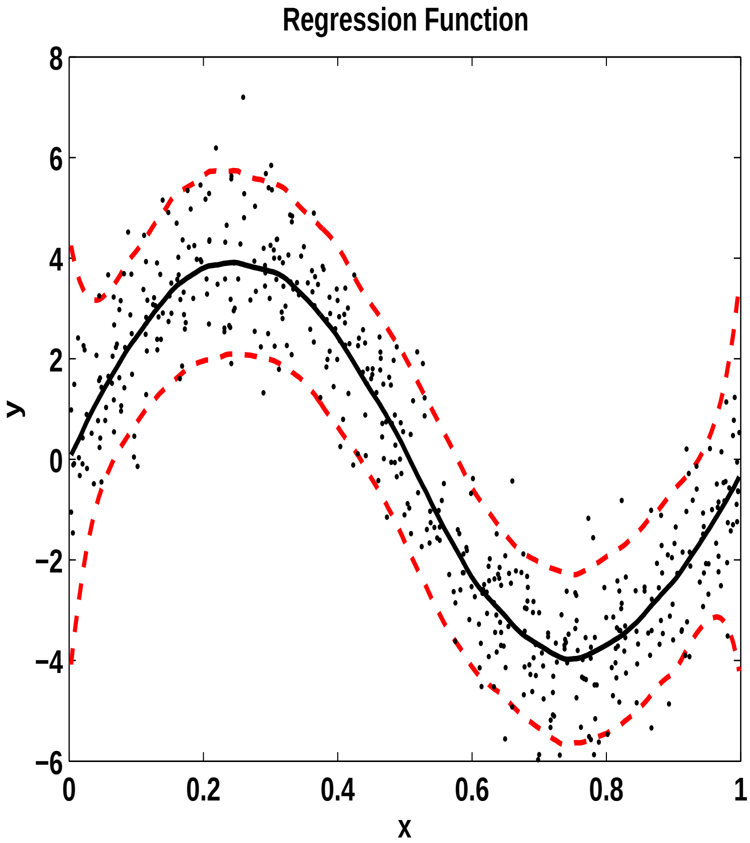


Figure 4.4. *Regression function and* 95% *confidence interval*

4.4 Simulation for Functional Linear Model

To test the quality for the functional linear model, we consider a simple case where there is only one predictor x ($p = 1$). $x_i, i = 1, \cdots, n$ are generated independently from $N(0, 1)$. Here, x is fixed, not a function of other parameters. There is no any additional technical difficulties in the estimating steps for the predictor x to be the function of u_j. $\varepsilon_{i,j}, i = 1, \cdots, n, j = 1, \cdots, n_i$ are also generated independently from $N(0, \sigma^2)$. The independent variable u is set to be evenly distributed within the interval $(0, 1)$, $u_j = \frac{j}{J+1}$, where $j = 1, \cdots, J$.

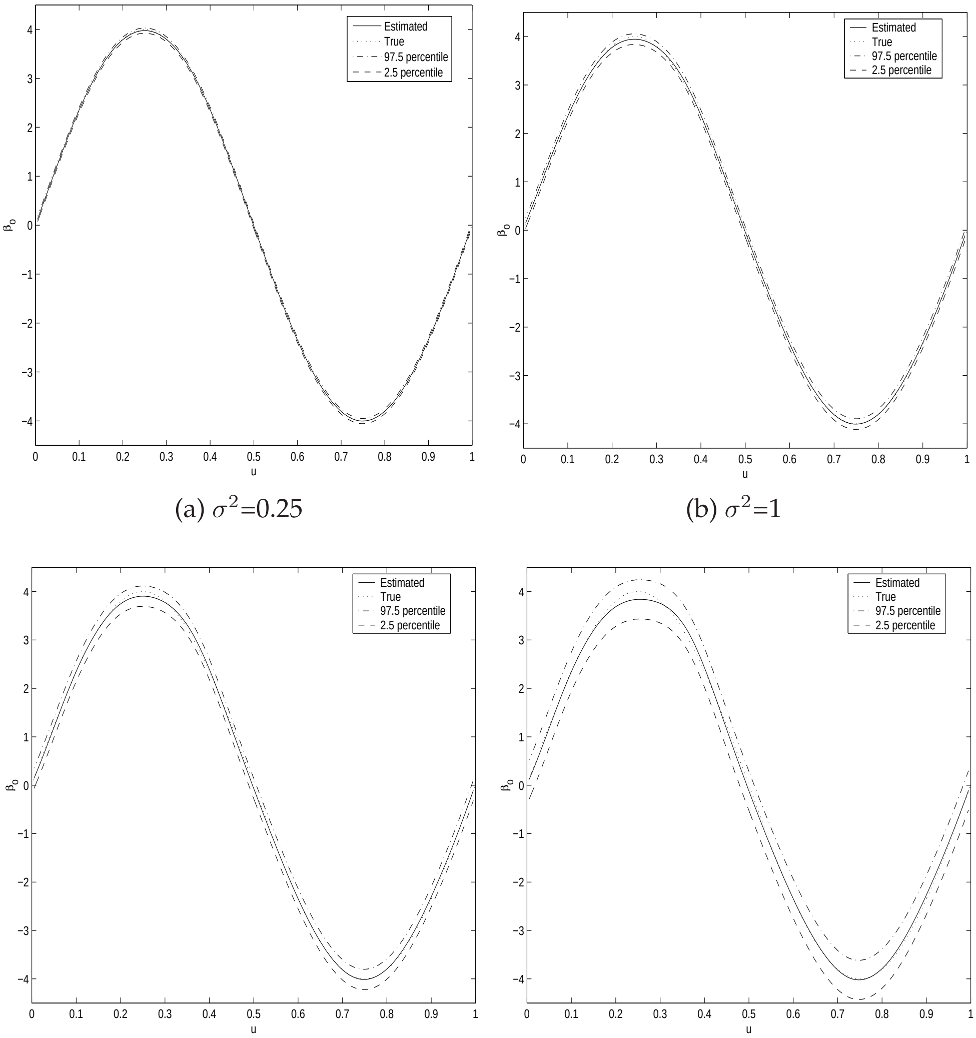
The random functional observations $y_i(u_j)$ are generated using $\beta_0(u) = 4\sin(2\pi u)$ and $\beta_1(u) = 16(u - 0.5)^2$. Then the functional linear model in this case turns into:

$$\begin{aligned} y_i(u_j) &= \beta_0(u_j) + \beta_1(u_j)x_i + \varepsilon_{ij} \\ &= 4\sin(2\pi u_j) + 16(u_j - 0.5)^2 x_i + \varepsilon_{ij} \end{aligned} \tag{4.19}$$

For this example, we choose to set $n = 100$, $J = 200$. Figure 4.5 shows the fitted value for the functional coefficient $\beta_0(u)$ and the corresponding 97.5 percent quantile and 2.5 percent quantile curves for different values of σ^2. We can see for small variance case, the estimation is very good and even for the largest variance case, the estimation is still very good.

Figure 4.6 shows the fitted value for the functional coefficient $\beta_1(u)$ and the corresponding 97.5 percent quantile and 2.5 percent quantile curves for different values of σ^2. We can see for small variance case, the estimation is very good for this quadratic functional coefficient. Even in the largest variance case, the estimation is still very good for this quadratic functional coefficient.

To address the importance of the second smoothing steps in terms of the MSE for the functional coefficients, we consider the following setup: $N = 5000$ groups of the different random sample of the same size are generated and the above func-



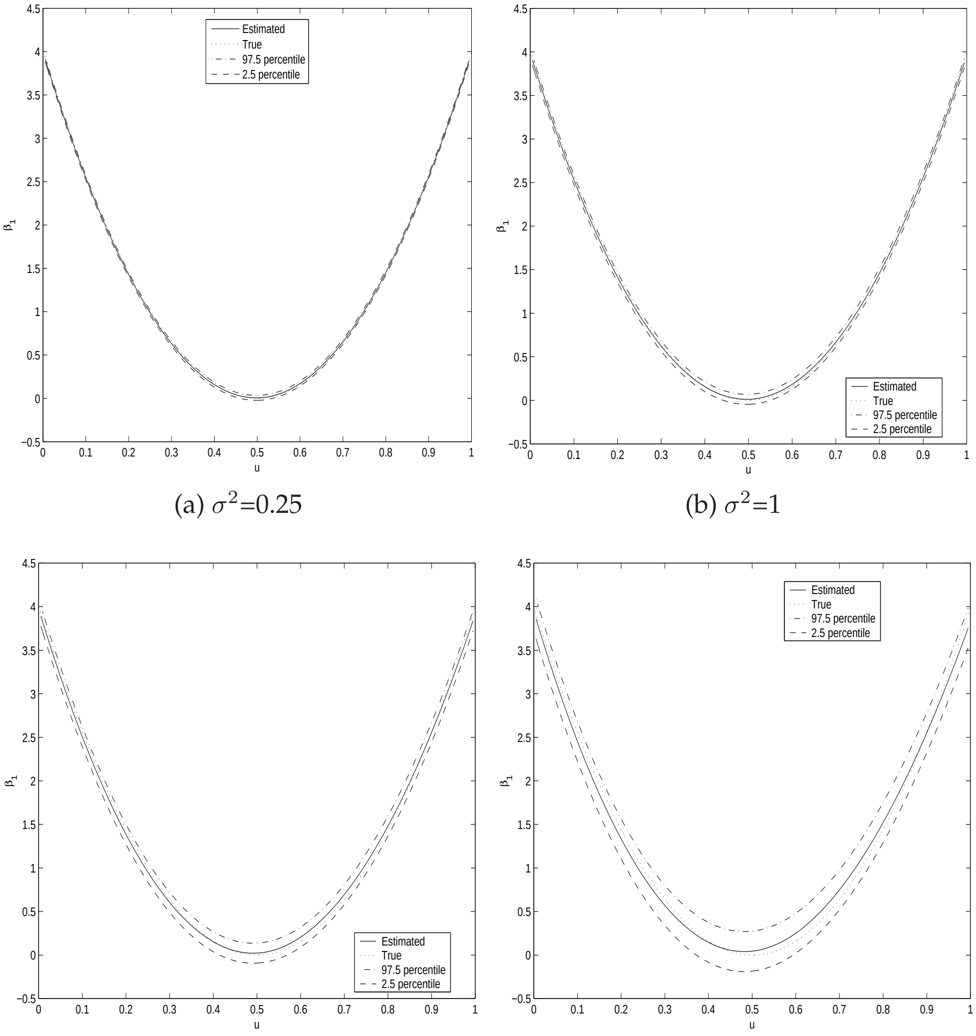
(a) σ^2=0.25

(b) σ^2=1

(c) σ^2=2

(d) σ^2=4

Figure 4.5. *Estimates of functional coefficient $\boldsymbol{\beta}_0$ for model (4.19). Solid line is the estimated value, dotted line is the true value.*



(a) σ^2=0.25

(b) σ^2=1

(c) σ^2=2

(d) σ^2=4

Figure 4.6. *Estimates of functional coefficient $\boldsymbol{\beta}_1$ for model (4.19). Solid line is the estimated value, dotted line is the true value.*

Table 4.3. *MMSE and VarMSE*

j	$i = 0(\beta_0(s.e.))$	$i = 1(\beta_1(s.e.))$
$1(\tilde{\beta})$	0.0204(3.35e-03)	0.0214(5.62e-03)
$2(\hat{\beta})$	0.0019(1.10e-03)	0.0014(1.11e-03)

tional linear regression steps are iterated for N times. For each random sample group k, the MSEs for both the estimates from the 1st step and 2nd step ($MSE_i^{(1)}(k)$ and $MSE_i^{(2)}(k)$) are calculated. Also the means and variances of these MSEs are calculated. These are defined by:

$$MMSE_i^{(j)} = \frac{1}{N}\sum_{l=1}^{N} MSE_i^{(j)}(l)$$

for the means of MSEs and:

$$VarMSE_i^{(j)} = \frac{1}{N-1}\sum_{l=1}^{N}(MSE_i^{(j)}(l) - MMSE_i^{(j)})^2$$

for the sample variances of MSEs. Tables 4.3 shows the values for $MMSE_i^{(j)}$ and $VarMSE_i^{(j)}$. From these two tables, we can see the MSEs for the smoothed parameters is only a fraction of the unsmoothed ones. We observe the same situation for the variance of the MSEs. This shows the importance for the smoothing step.

Another way to look at the benefit of the smoothing step is to look at the ratio of the two MSEs for both step of each group. Let:

$$r_i(k) = \frac{MSE_i^1(k)}{MSE_i^2(k)}$$

where $i = 0$ is for $\beta_0(u)$ and $i = 1$ is for $\beta_1(u)$. $r_i(k)$ will show the benefit we can get from the smoothing step. The larger this ratio is, the more gain we get from penalized spline regression step. Figure 4.7 gives the boxplot for these two log ratios because of the wide range of the ratios. From this plot, we can see that most of the ratios is larger than 10. So the smoothing step is very efficient.

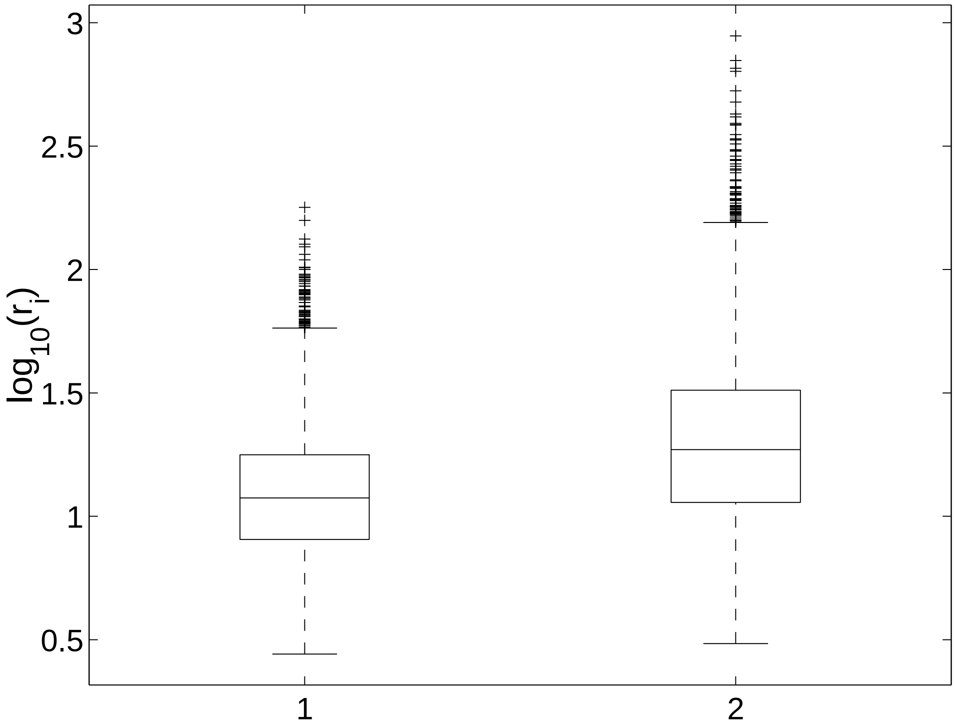


Figure 4.7. *Boxplot of the log ratio MSEs.*

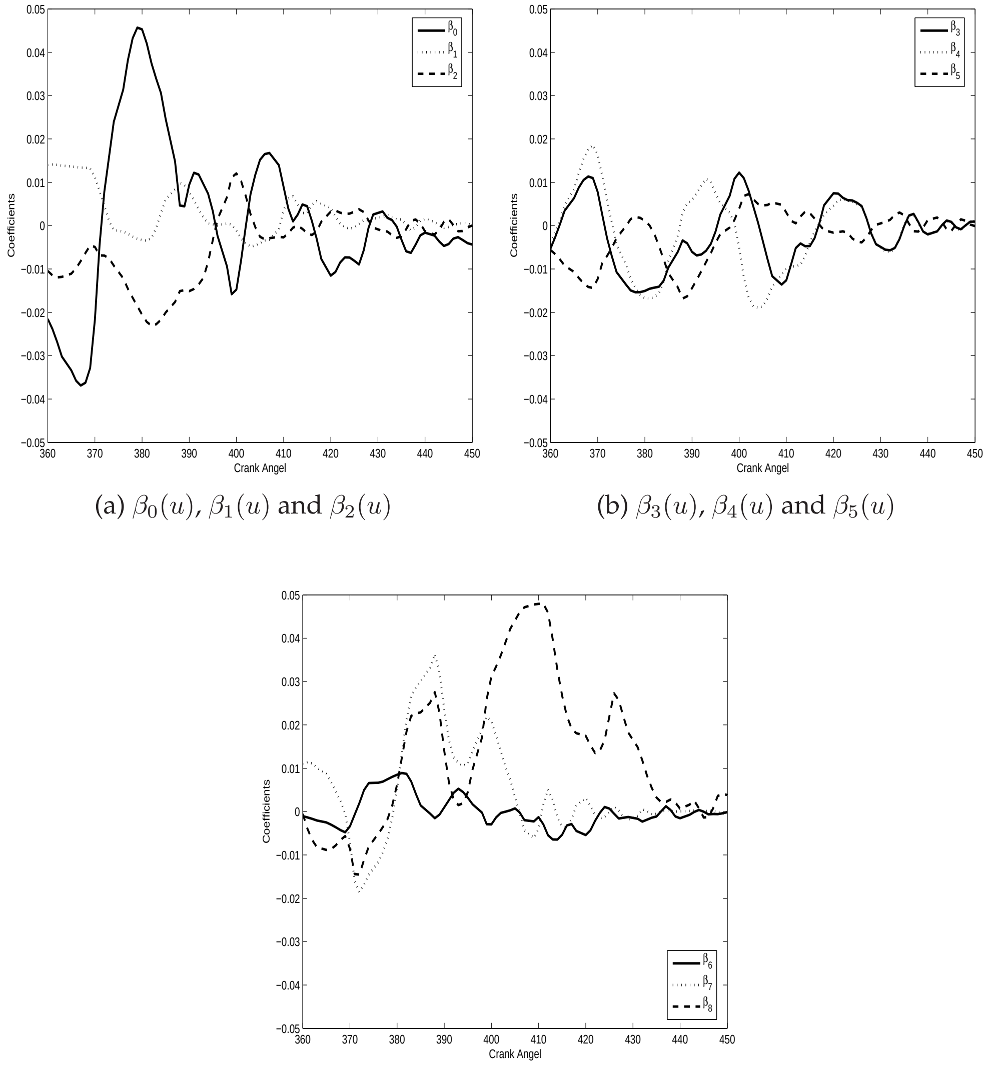
4.5 Case Study: Mean function estimation for valve motion error for crank angle within $(360, 450)$ using FLM

In this section, part of the valvetrain data (the crank angle within (360,450)) will be analyzed by functional linear model using the two-step estimation procedure discussed in previous sections.

The ordinary least square estimate $\hat{\boldsymbol{\beta}}(u)$ for the model:

$$y_i(u) = \mathbf{X}_i\boldsymbol{\beta}(u) + \epsilon$$

are given in Figure 4.8 . Note that there are eight predictors and the number of functional coefficient is nine including one baseline function $\beta_0(u)$. From these figures, we can see all the coefficients have large variation along with the crank angel. So the function linear model is suitable for this part of the problem. Figures 4.9 to 4.10



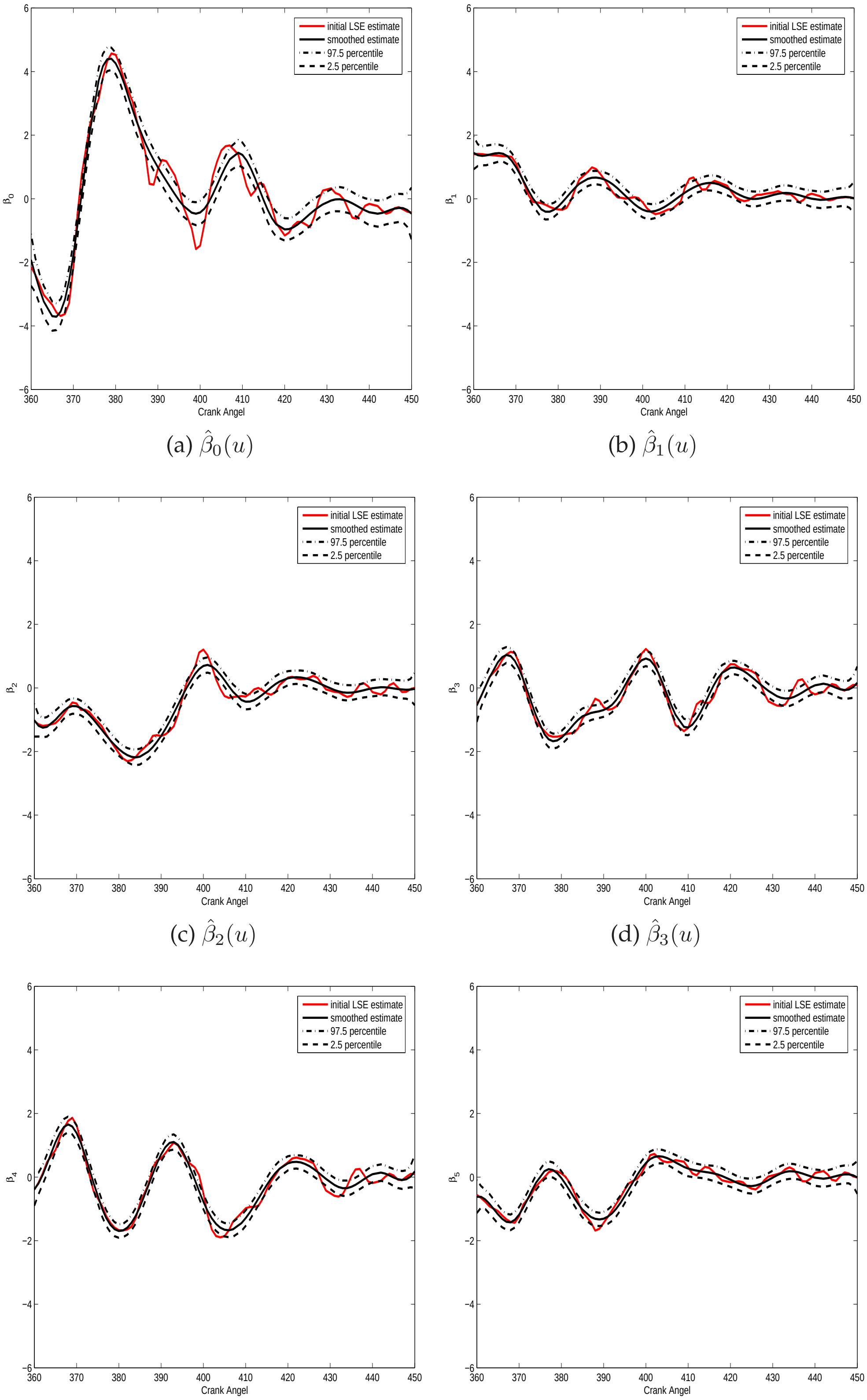
(a) $\beta_0(u)$, $\beta_1(u)$ and $\beta_2(u)$

(b) $\beta_3(u)$, $\beta_4(u)$ and $\beta_5(u)$

(c) $\beta_6(u)$, $\beta_7(u)$ and $\beta_8(u)$

Figure 4.8. *OLS estimates for the functional coefficients within crank angle region (360,450)*

give the smoothed functional coefficients together with the 97.5 percentile and 2.5 percentile curves from the functional linear model using the procedure described in chapter 4. The result is very good. Figures 4.11 to 4.13 give the FLM fitted value, OLS fitted value and the SAE simulation result comparison for all the 16 design cases. From these plots, we can see most of the FLM fitted results are very good but for some design cases, the smoothing of functional coefficients along crank angle direction distorts the original information with a certain degree.

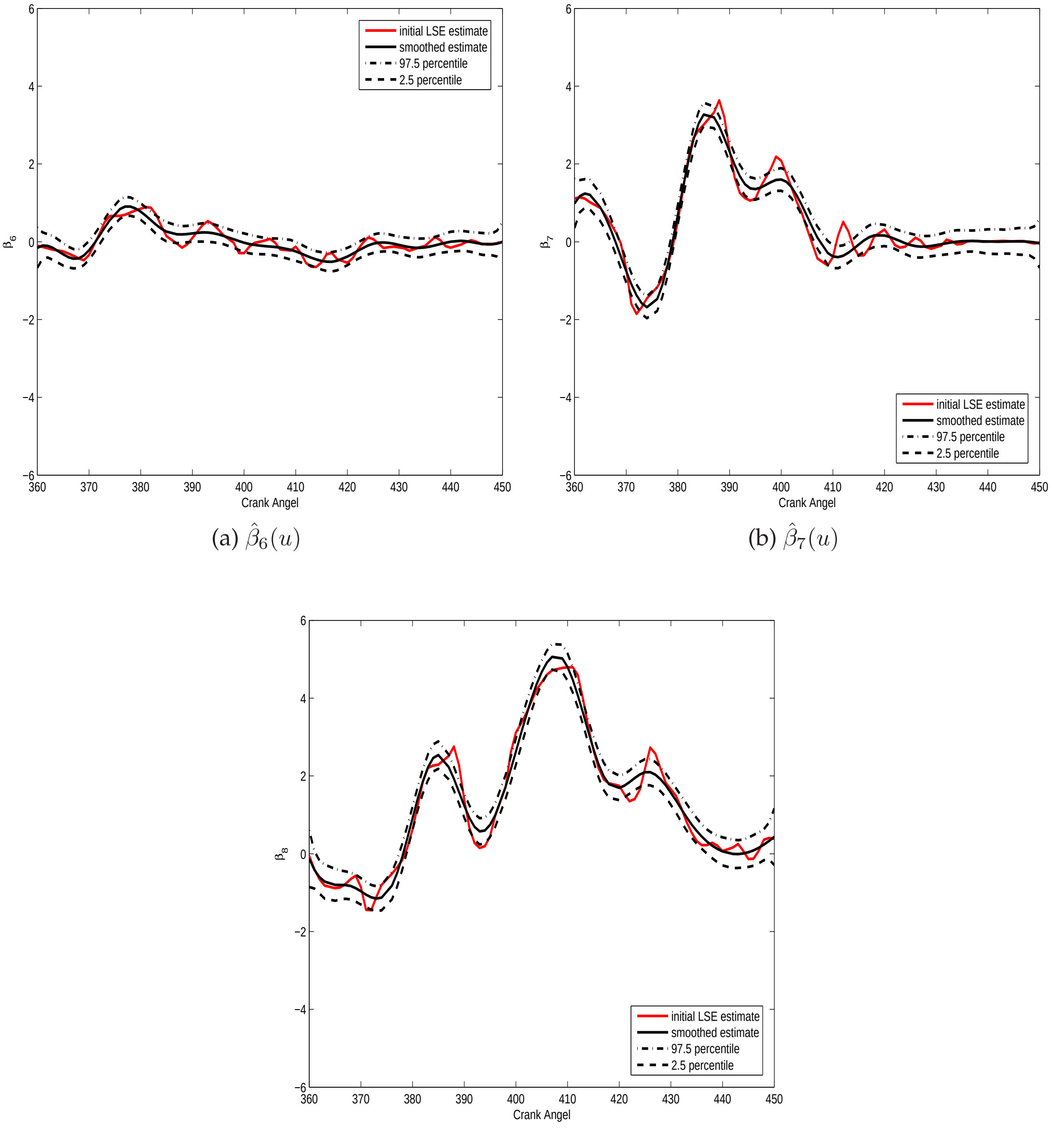


(a) $\hat{\beta}_0(u)$ (b) $\hat{\beta}_1(u)$

(c) $\hat{\beta}_2(u)$ (d) $\hat{\beta}_3(u)$

(e) $\hat{\beta}_4(u)$ (f) $\hat{\beta}_5(u)$

Figure 4.9. *Estimated coefficients from functional linear model within crank angel region (360,450) with the design matrix in Table 4.1*

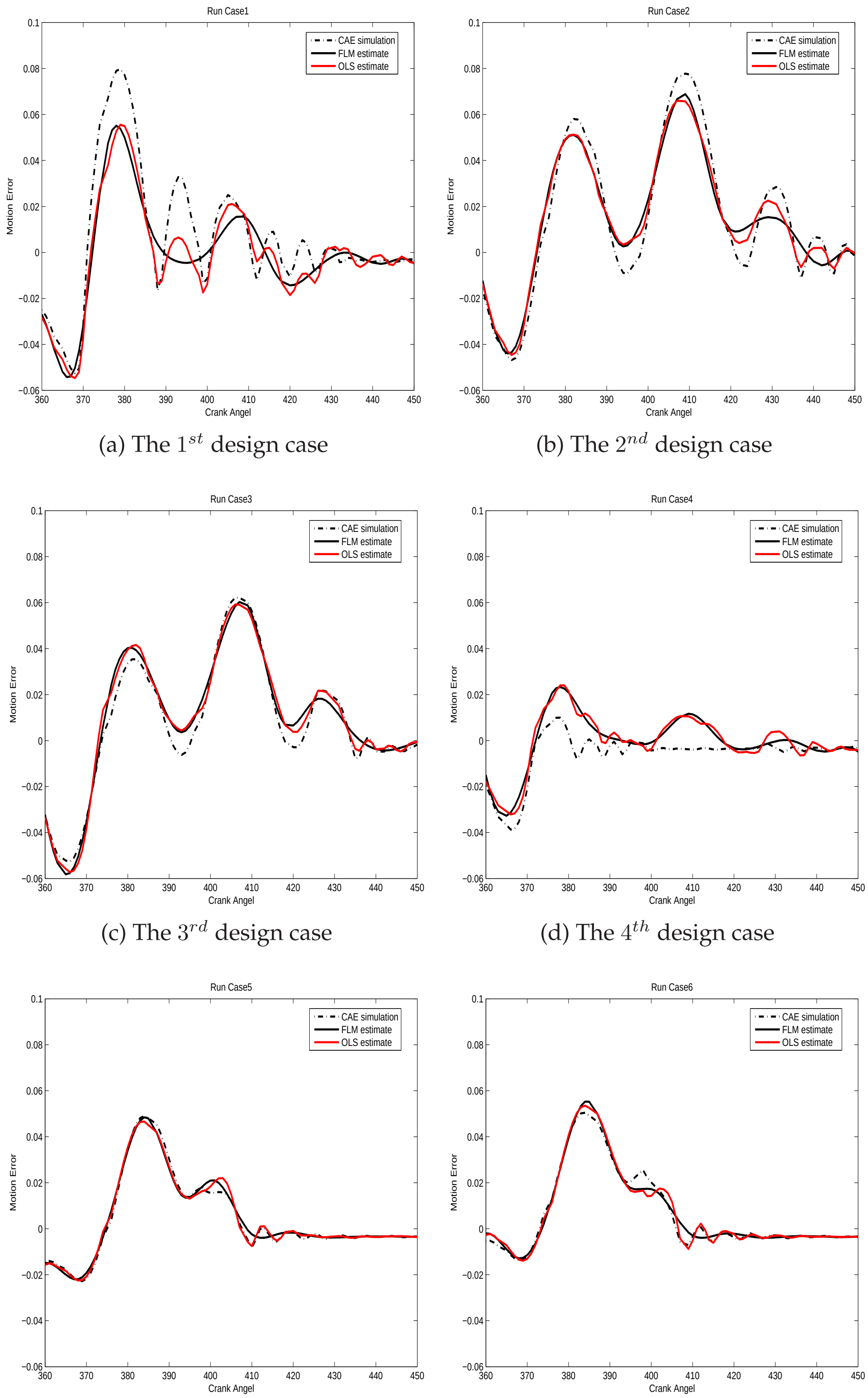


(a) $\hat{\beta}_6(u)$

(b) $\hat{\beta}_7(u)$

(c) $\hat{\beta}_8(u)$

Figure 4.10. *LSE for the functional coefficients within crank angle region (360,450) with the design matrix in Table 4.1*



(a) The 1^{st} design case

(b) The 2^{nd} design case

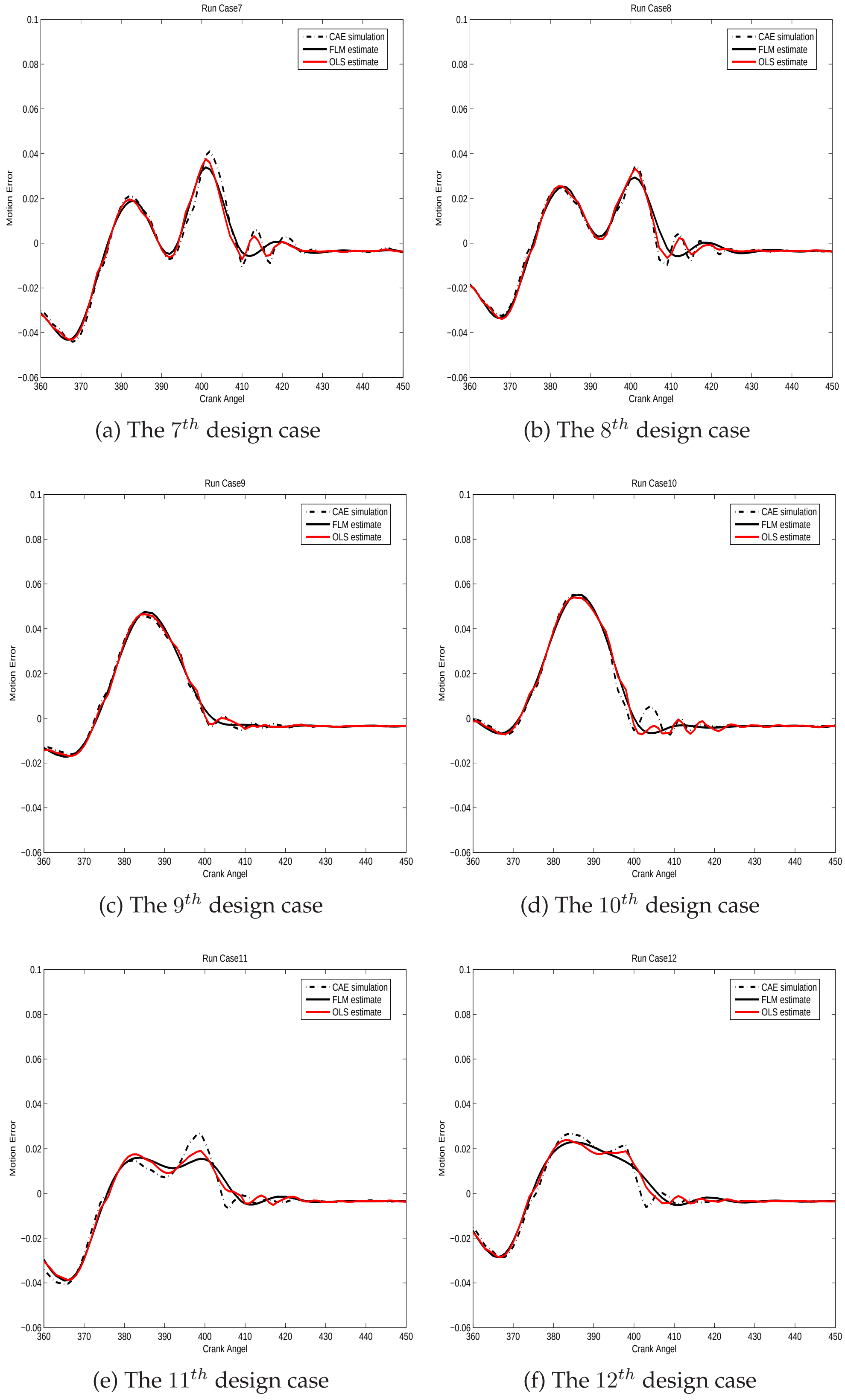
(c) The 3^{rd} design case

(d) The 4^{th} design case

(e) The 5^{th} design case

(f) The 6^{th} design case

Figure 4.11. *Valvetrain motion errors comparison for the first 6 design cases in Table 4.1 within crank angel region (360,450)*



(a) The 7^{th} design case

(b) The 8^{th} design case

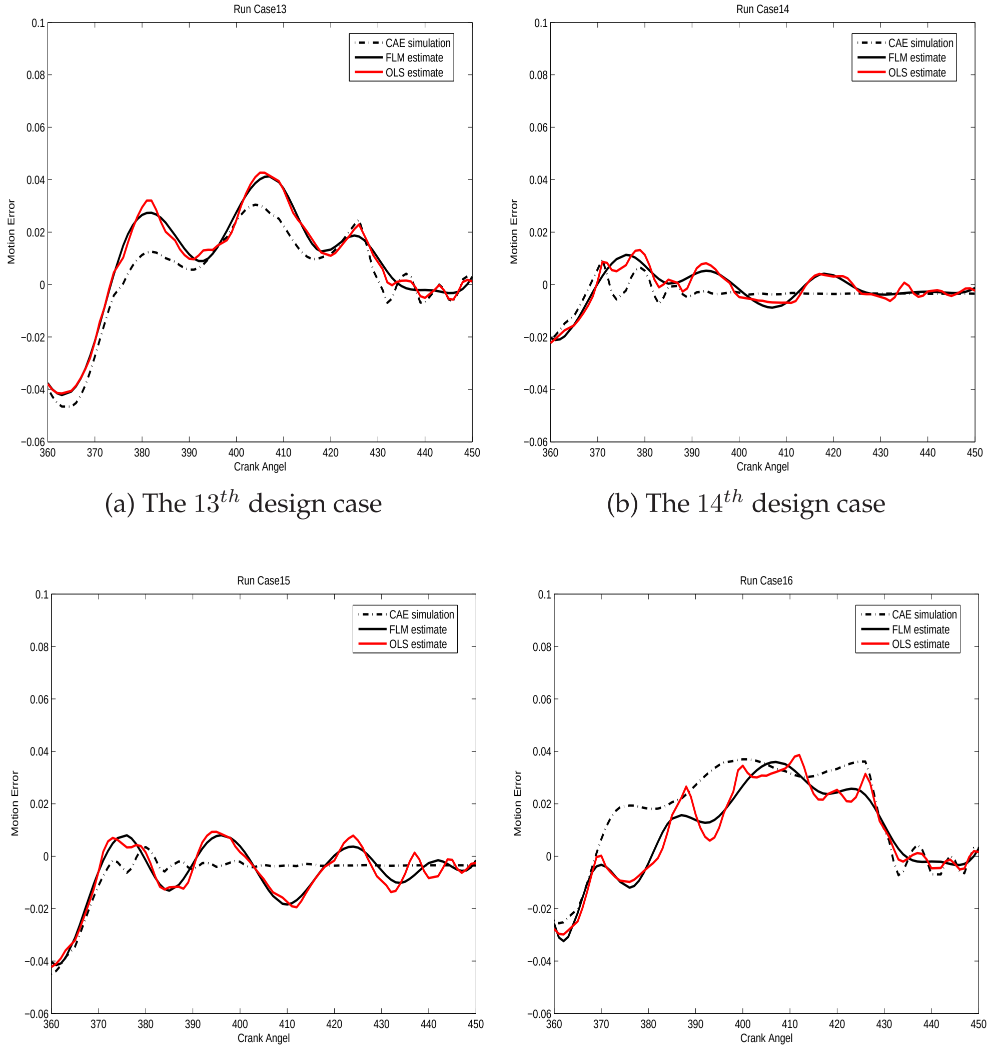
(c) The 9^{th} design case

(d) The 10^{th} design case

(e) The 11^{th} design case

(f) The 12^{th} design case

Figure 4.12. *Valvetrain motion errors comparison for the second 6 design cases in Table 4.1 within crank angel region (360,450)*



(a) The 13th design case

(b) The 14th design case

(c) The 15th design case

(d) The 16th design case

Figure 4.13. *Valvetrain motion errors comparison for the last 4 design cases in Table 4.1 within crank angel region (360,450)*

4.6 Functional ANOVA for Functional Linear Model with Kriging

Similar to the case in Chapter 3, we are interested in knowing which predictor has more effect on the response as well. We use functional ANOVA decomposition to analyze this problem.

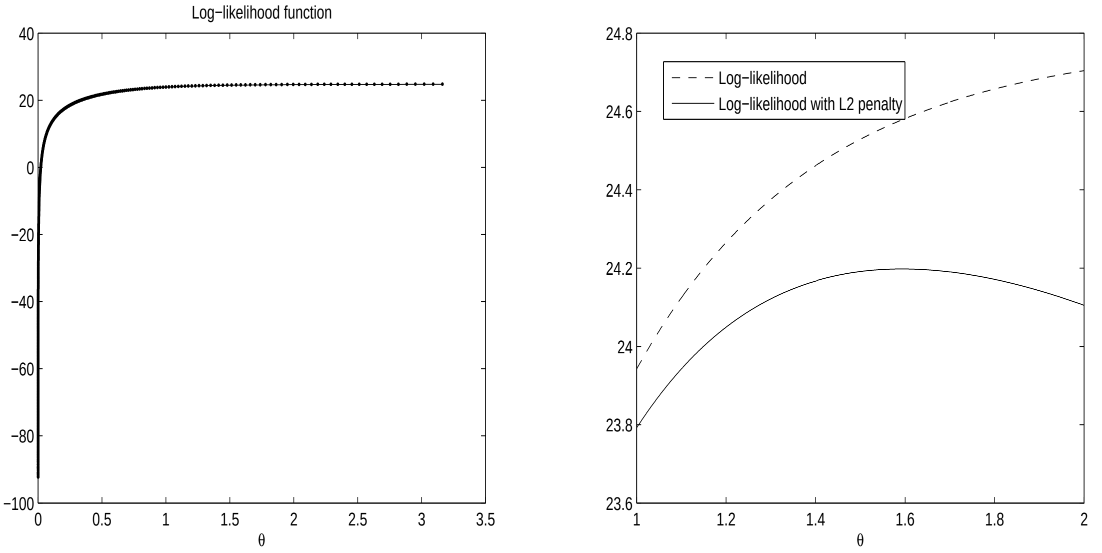
The functional linear model with kriging term can be written as:

$$m(\mathbf{x};t) = \mathbf{x}\boldsymbol{\beta}(t) + \mathbf{r}(\mathbf{x};t)\mathbf{R}^{-1}(\mathbf{Y}(t) - \mathbf{X}\boldsymbol{\beta}(t)), \tag{4.20}$$

where $\boldsymbol{\beta}(t)$ is the functional coefficient vector, $\mathbf{Y}(t)$ is the observed response vector at index t, $\mathbf{X}$ is the design matrix. From the last section, we know that the two-step kriging model with interpolation in temporal direction is a more favorable model and we still use this procedure to estimate the parameter in this model.

Here, we assume Gaussian correlation function with single parameter θ as the structure of correlation matrix at each index point. As shown in figure (4.14.(a)), the Newton-Ralphson algorithm will not work well on the original likelihood function since this function will flat out at higher range of θ. The penalized likelihood with $L2$ penalty is utilized to get the optimal θ and $\boldsymbol{\beta}$. In order to show the comparison of these two likelihoods, figure (4.14.(b)) zooms into the zone where the penalized likelihood has the maximum point. From this plot, we can see clearly the maximum point after the penalty.

The parameter estimation is performed as follows. At each functional index point, θ value is searched on a pre-defined dense grid point set. For each θ value, the corresponding weighted least square estimate for $\boldsymbol{\beta}$ is computed followed by the calculation of penalized likelihood. The MLE for θ (and $\boldsymbol{\beta}$) is then picked which corresponding to the maximum likelihood value. After all the θ estimation is ready along the functional direction, a non-parametric smoothing is performed to get



(a) Log-likelihood

(b) Log-likelihood with $L2$ penalty

Figure 4.14. *Log-likelihood function for FLM with kriging*

the smoothed version of θ. The coefficient $\boldsymbol{\beta}$ is then calculated again by using the weighted least square estimation.

After the estimation of $\hat{\theta}$ and $\hat{\boldsymbol{\beta}}$, we can directly plug them into (4.20) to get the function which needs to be decomposed orthogonally onto each variable direction by functional ANOVA:

$$g(\mathbf{x};t) = g_0(t) + \sum_{i=1}^{d} g_i(x_i;t) + \sum_{i<j} g_{ij}(x_i, x_j;t) + \cdots + g_{1\cdots d}(x_1, \cdots, x_d;t). \qquad (4.21)$$

This functional ANOVA decomposition at each functional point is a direct application of the work of Sobol′ (1993), Sobol′ (2001) and Sobol′ (2003). Please refer to Chapter 3 for the properties of this decomposition.

Denote the total variance as

$$D(t) = \mathrm{Var}g(t) = E(g(t)^2) - (Eg(t))^2, \qquad (4.22)$$

and the component variance as:

$$D_{i_1\cdots i_k}(t) = \text{Var}g_{i_1\cdots i_k} = \int g^2_{i_1\cdots i_k}(x_{i_1}, \cdots, x_{i_k})dx_{i_1}\cdots dx_{i_k}. \tag{4.23}$$

Due to the orthogonality, we have:

$$D(t) = \sum_{k=1}^{d} \sum_{i_1<\cdots<i_k} D_{i_1\cdots i_k}(t).$$

Then the Sobol′ indices are:

$$S_{i_1\cdots i_k} = \frac{D_{i_1\cdots i_k}}{D}.$$

These indices reflect how much does each component in decomposition (4.21) affect the response.

In order to calculate the functional decomposition conveniently, we normalize the predictor $\mathbf{x} \in (0,1)^d$ first. Denote:

$$\mathbf{M} = (M_1, M_2, \cdots, M_n)^T = \mathbf{R}^{-1}(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}), \tag{4.24}$$

(4.20) turns into:

$$g(t) = \hat{\beta}_0(t) + \sum_{i=1}^{d} x_i\hat{\beta}_i(t) + \sum_{i=1}^{n} r_i(t)M_i(t). \tag{4.25}$$

The overall mean effect turns into:

$$g_0(t) = Eg(t) = \int g(t)d\mathbf{x} = \hat{\beta}_0(t) + \frac{1}{2}\sum_{i=1}^{d}\hat{\beta}_i(t) + \sum_{i=1}^{n} M_i(t)\int r_i(t)d\mathbf{x}. \tag{4.26}$$

The linear component turns into:

$$g_k(x_k, t) = \int g(t)\Pi_{i\neq k}dx_i - Eg = (x_k - \frac{1}{2})\hat{\beta}_i(t) + \sum_{i=1}^{n} M_i \int r_i\Pi_{l\neq k}dx_l - \sum_{i=1}^{n} M_i \int r_i d\mathbf{x}. \tag{4.27}$$

In order to get the overall variance, we need to get $Eg^2(t)$, which can be calculated in the following way:

$$\begin{aligned} Eg^2 &= \textstyle\int_{\mathbf{x}\in[0,1]^d} g^2 d\mathbf{x} \\ &= \textstyle\hat{\beta}_0^2 + \hat{\beta}_0\sum_{j=1}^{d}\hat{\beta}_j + 2\hat{\beta}_0\sum_{i=1}^{n} M_i\int_{\mathbf{x}\in[0,1]^d} r_i d\mathbf{x} + \frac{1}{3}\sum_{j=1}^{d}\hat{\beta}_j^2 \\ &\quad \textstyle+\frac{1}{2}\sum_{j=1}^{d}\sum_{k=j+1}^{d}\hat{\beta}_j\hat{\beta}_k + \sum_{i=1}^{n}\sum_{j=1}^{n} M_iM_j\int_{\mathbf{x}\in[0,1]^d} r_ir_j d\mathbf{x} \\ &\quad \textstyle+\sum_{i=1}^{n}\sum_{j=1}^{d}\hat{\beta}_j\int r_i\Pi_{k\neq j}dx_k\int_0^1 x_j exp(-\hat{\theta}(x_j - X_{i,j})^2)dx_j. \end{aligned} \tag{4.28}$$

The integration terms like $\int r_i d\mathbf{x}$ and $\int r_i r_j d\mathbf{x}$ are given in (3.64) and (3.66).

4.7 Simulation–Coefficients Estimates and Functional ANOVA

Our purpose in this section is to validate the code for parameter estimation and functional ANOVA decomposition for model (4.20). We pick the same design matrix as in table 4.1, which is the same design matrix as in Example 1.2. Let the true functional coefficient $\boldsymbol{\beta}(t)$ to be the estimated functional coefficient in model (4.20) using the data from Example 1.2. The error term needs to be smooth, which is required by the kriging model. We pick the following formula for the error term:

$$\epsilon_{ij} = c[\sin(2k\pi t_j) + \cos(2k\pi t_j)]exp(-\frac{1}{2}\mathbf{x}_i\mathbf{A}_j\mathbf{x}_i'), \tag{4.29}$$

where $i = 1, 2, \cdots, 16$, $j = 1, \cdots, 83$. $\mathbf{A}_j$ is a square matrix with off diagonal elements equal to ρ_j, and diagonal element equal to unit. ρ_j is uniformly randomly picked within range $(0.3, 0.7)$. $\mathbf{A}_j$ can be viewed as the correlation matrix for $\mathbf{x}_i$ at the $j-th$ functional index. In this simulation study, we take $k = 12$. c is a constant which controls the noise-signal ratio. In order to see the effect to the code by the noise level, it is taken to be the following values: 0.01,0.1,0.5,1,2,5,10. The true model for this simulation is:

$$y_{ij} = \mathbf{x}_i\boldsymbol{\beta}_j + \epsilon_{ij}. \tag{4.30}$$

To investigate the sensitivity of the code to the noise ratio, we calculate the mean-square-error (MSE) between true value and simulation result for functional coefficients $\boldsymbol{\beta}_i$ $(i = 0, 1, \cdots, 8)$ and individual variance component in functional ANOVA. The MSE for coefficient is:

$$MSE_i(c) = \frac{1}{n}\sum_{j=1}^{83}(\beta_{i,j} - \hat{\beta}_{i,j})^2, \tag{4.31}$$

where $n = 83$, $\beta_{i,j}$ is the true coefficient value for the i-th component at j-th functional index, $\hat{\beta}_{i,j}$ is the corresponding estimates. Apparently, this MSE is a function of noise

Table 4.4. *MSE for coefficients for FLM with kriging*

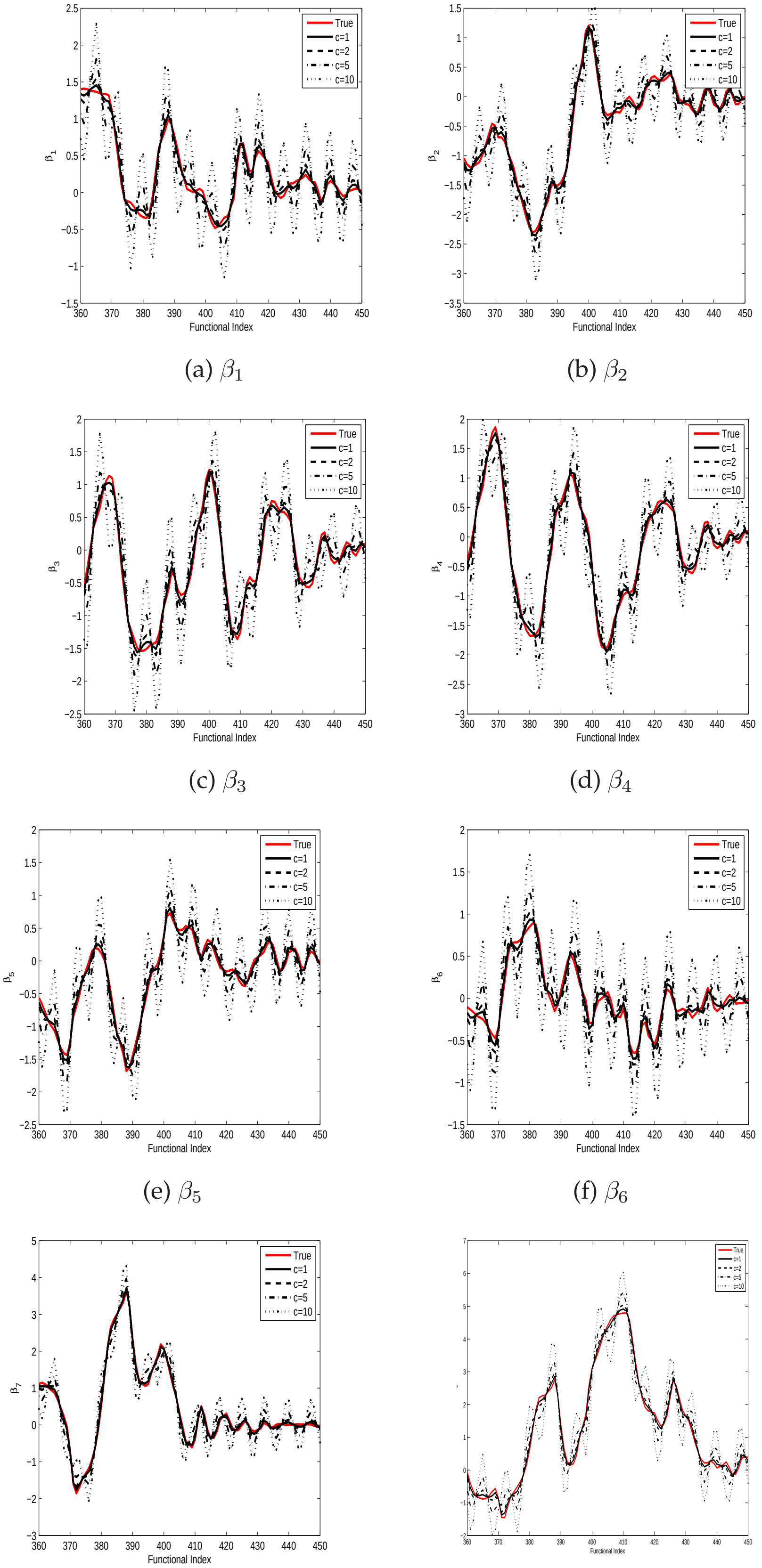
	$c = 0.01$	$c = 0.1$	$c = 0.2$	$c = 1$	$c = 2$	$c = 5$	$c = 10$
β_0	0.0000092	0.0009417	0.0223609	0.0952003	0.3754671	2.4088899	9.5717694
β_1	0.0000003	0.0000337	0.0008068	0.0034130	0.0134650	0.0863404	0.3427715
β_2	0.0000003	0.0000337	0.0008068	0.0034130	0.0134650	0.0863404	0.3427715
β_3	0.0000004	0.0000484	0.0011464	0.0049066	0.0193179	0.1244280	0.4936032
β_4	0.0000004	0.0000484	0.0011464	0.0049066	0.0193179	0.1244280	0.4936032
β_5	0.0000003	0.0000337	0.0008068	0.0034130	0.0134650	0.0863404	0.3427715
β_6	0.0000003	0.0000337	0.0008068	0.0034130	0.0134650	0.0863404	0.3427715
β_7	0.0000003	0.0000337	0.0008068	0.0034130	0.0134650	0.0863404	0.3427715
β_8	0.0000008	0.0000850	0.0020744	0.0085856	0.0339740	0.2165106	0.8594909

level c. Table 4.4 shows this MSE for $c = 0.01, 0.1, 0.2, 1, 2, 5$ and 10. We can see that the MSE will increase along with the increasing of the noise ratio. Within all the components, β_0 is most sensitive to the noise ratio c. Figure 4.15 shows the true and simulated coefficients under several different noise ratio situations. Figure 4.16 shows simulated β_0. From these figures, we can see that the model estimation for mean function will perform well if the noise ratio is not too large ($c \leq 2$). For large noise/signal ratio, the mean functional estimation will be distorted.

To investigate the sensitivity of functional ANOVA decomposition to the noise level, we calculate the following MSE:

$$MSE_i(c) = \frac{1}{n}\sum_{j=1}^{n}(\text{Var}g_{i,j} - \hat{\text{Var}}g_{i,j})^2, \tag{4.32}$$

where $n = 83$, $\text{Var}g_{i,j}$ is the variance component on x_i at functional index j. Table 4.5 gives this MSE for each variance component. Please note that the real MSE value equals to the value in this table divided by 10^4. Figures 4.17 and 4.18 show the estimated variance components and the corresponding true values for $c = 2$ and



(a) β_1

(b) β_2

(c) β_3

(d) β_4

(e) β_5

(f) β_6

(g) β_7

(h) β_8

Figure 4.15. *Simulation result for coefficient estimates for FLM with kriging*

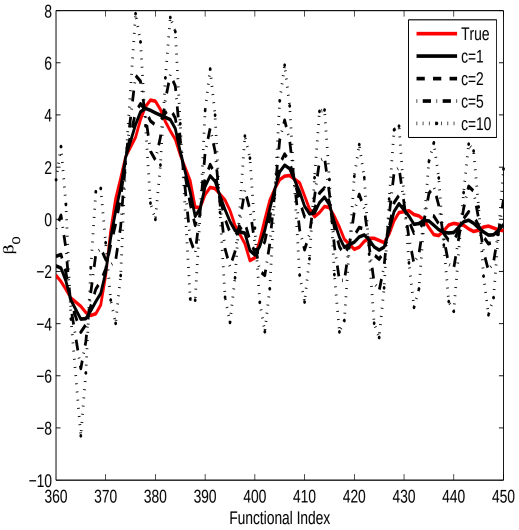
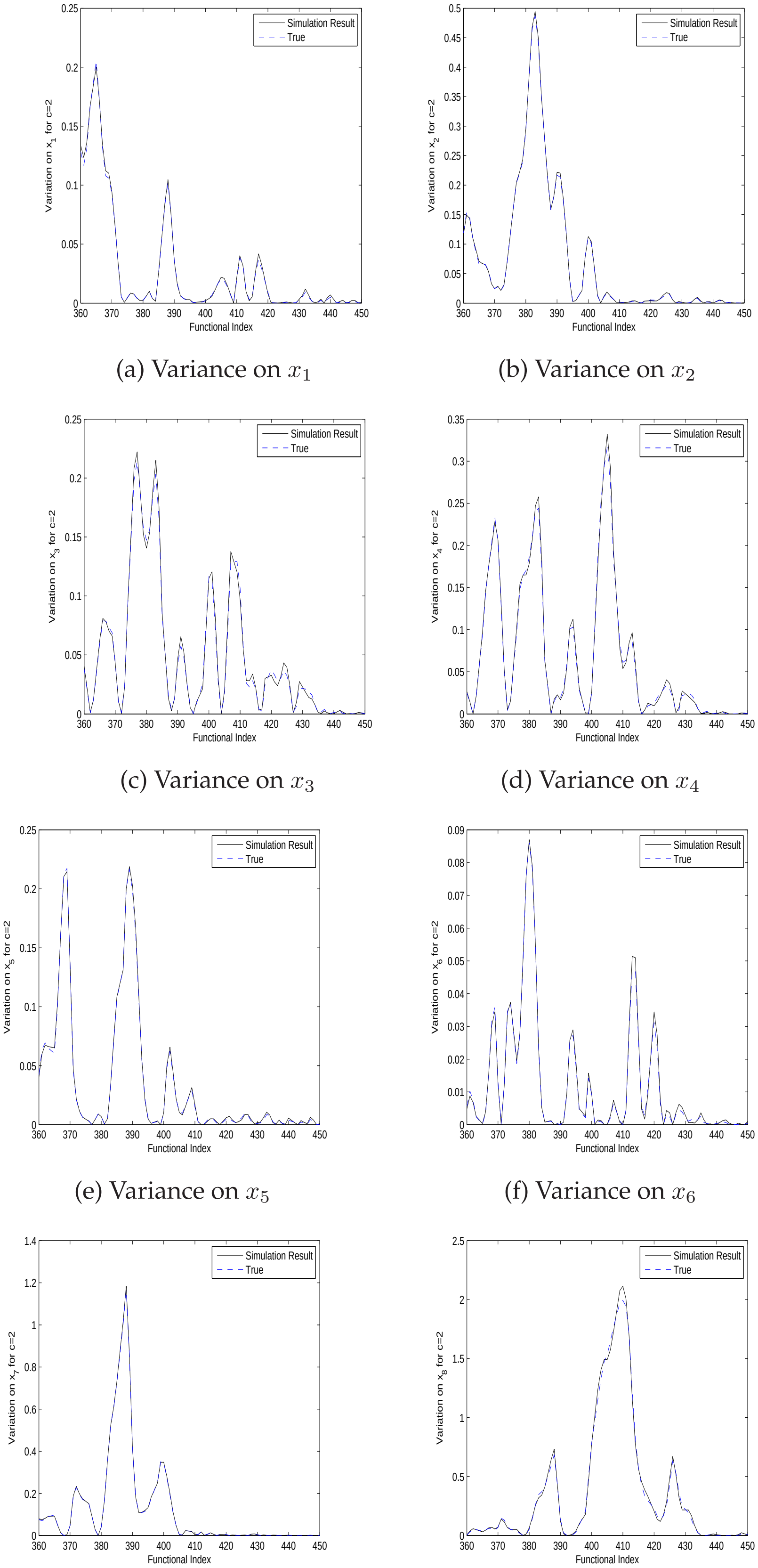


Figure 4.16. *Estimated β_0 for FLM with kriging*

$c = 5$. Figure 4.19 shows the estimated overall variance and true value under different noise ratio situations. From these results, we can see that from MSE point of view, the functional ANOVA decomposition is less sensitive to the noise ratio compared to the mean function estimation.

4.8 Simulation–Comparison of Multivariate and Single Response

In this section, we make a comparison between multivariate kriging model and kriging model with single response. Equation (4.30) is used to generate the simulation response on the given design matrix in table 4.1. Furthermore, we generate another N random observations for the same predictors in table 4.1, where $N = 1000$. Using FLM, we obtain the main functions estimates and get the residual. We label these points as $\mathbf{x}_i^*$, where $i = 1, \cdots, N$, and the corresponding functional residual as $\mathbf{y}_i^*$. Then we use kriging with single response and multivariate kriging model to build models on design matrix separately. Then the predictions on each of the ran-



(a) Variance on x_1

(b) Variance on x_2

(c) Variance on x_3

(d) Variance on x_4

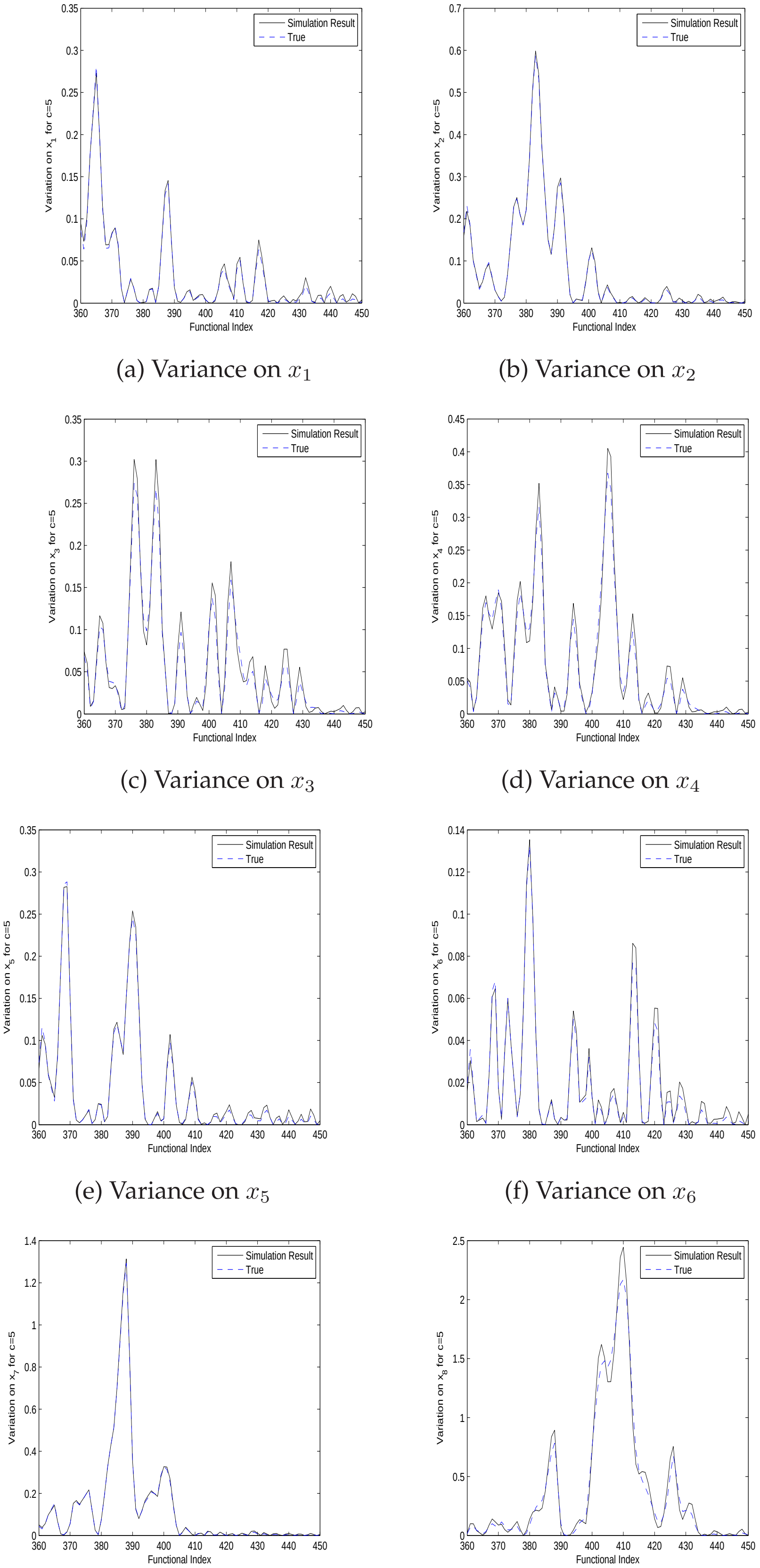
(e) Variance on x_5

(f) Variance on x_6

(g) Variance on x_7

(h) Variance on x_8

Figure 4.17. *Variance on each components for $c = 2$ for FLM with kriging*



(a) Variance on x_1

(b) Variance on x_2

(c) Variance on x_3

(d) Variance on x_4

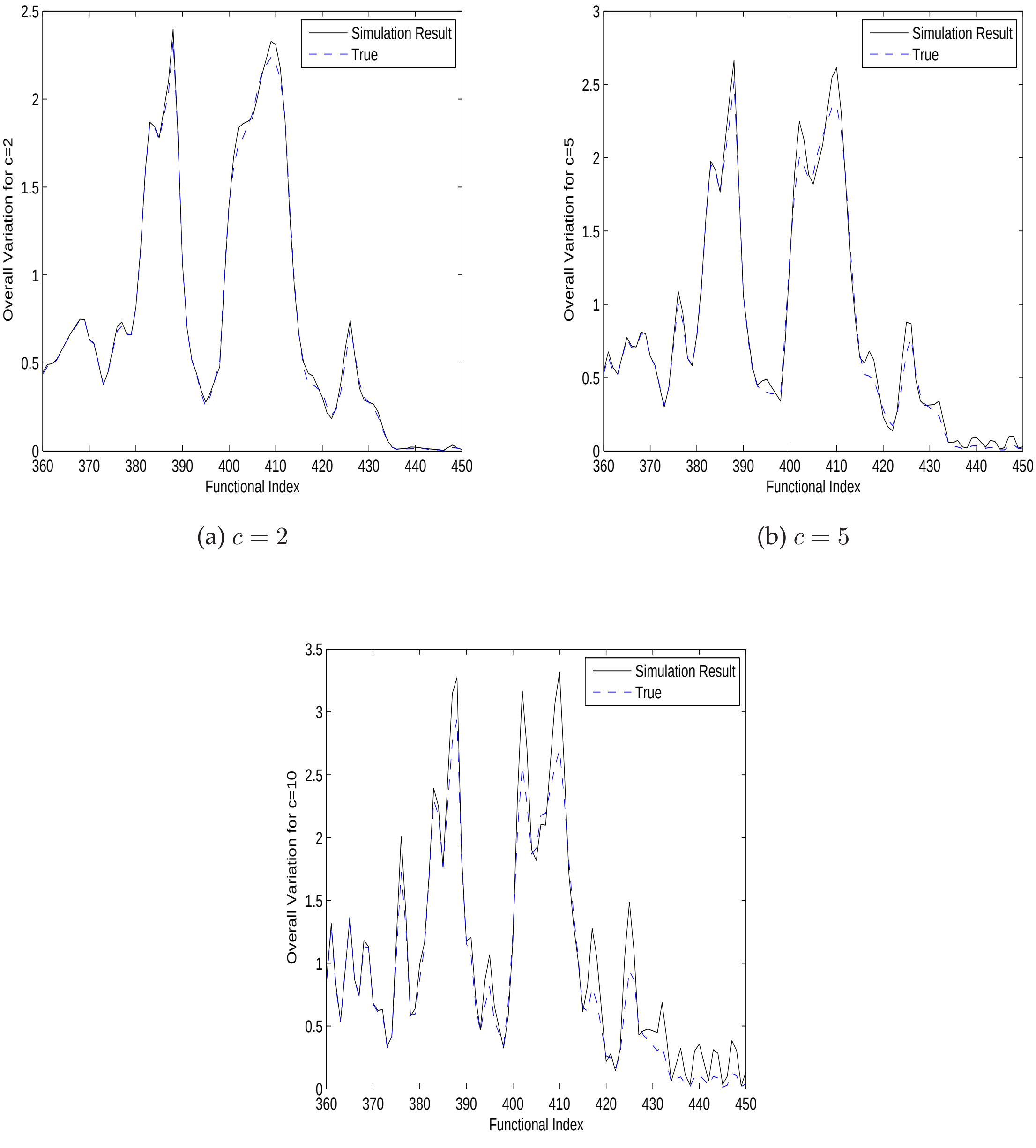
(e) Variance on x_5

(f) Variance on x_6

(g) Variance on x_7

(h) Variance on x_8

Figure 4.18. *Variance on each components for $c = 5$ for FLM with kriging*



(a) $c = 2$

(b) $c = 5$

(c) $c = 10$

Figure 4.19. *Overall variance comparison*

Table 4.5. *MSE for variance of individual components*

$MSE * 10^4$	$c = 0.01$	$c = 0.1$	$c = 0.2$	$c = 1$	$c = 2$	$c = 5$	$c = 10$
Var$g_1(x_1)$	0.0030	0.0033	0.0052	0.0112	0.0338	0.1350	1.4204
Var$g_2(x_2)$	0.0172	0.0173	0.0205	0.0234	0.0520	0.1871	1.6312
Var$g_3(x_3)$	0.0052	0.0053	0.0182	0.0503	0.1985	1.4023	8.6714
Var$g_4(x_4)$	0.0122	0.0123	0.0301	0.0765	0.2910	1.9507	10.8222
Var$g_5(x_5)$	0.0040	0.0040	0.0051	0.0080	0.0269	0.1389	1.5303
Var$g_6(x_6)$	0.0004	0.0004	0.0013	0.0029	0.0126	0.0931	1.2444
Var$g_7(x_7)$	0.0631	0.0636	0.0692	0.0797	0.1165	0.2947	1.7435
Var$g_8(x_8)$	0.4201	0.4457	1.1185	2.9694	10.6880	60.4602	274.0258
Overall Variance	2.1780e-04	1.9903e-02	0.6147	2.3921	13.0000	230.0844	3280.22

domly generated points are obtained from each model. The prediction errors for each functional index are defined as

$$P.E.(j) = \sum_{i=1}^{N} \|(y^*(j)_i - \hat{y}^*(j)_i)\|^2,$$

where $y^*(j)$ is the ith residual at functional index j, $\hat{y}^*(j)$ is the predicted value.

We use 2 points in functional direction for multivariate version to compare with the single response version. The result shows that the prediction error of single version is uniformly greater than the one of multivariate version. The averaged prediction error across functional direction for single version is 22.0001 for $c = 10$ (the magnitude for the error term), while the corresponding result for multivariate version is 20.1894. Table 4.6 shows the detailed comparison for some functional index j, where gain is defined as $\frac{P.E.(single) - P.E.(multivariate)}{P.E.(single)}$. From this table, we can see we gain a lot by using multivariate kriging model compared to kriging with single response. Also note that for different functional index, we have difference gains. This clearly shows the advantage of multivariate kriging model.

Table 4.6. *Comparison for multivariate kriging and kriging with single response in FLM for* $c = 10$

$Source$	$j=1$	$j=3$	$j=5$	$j=7$	$j=9$	$j=11$	$j=13$
$P.E.(single)$	23.1115	18.0262	45.2933	5.7887	35.0812	13.5569	25.8520
$P.E.(multivariate)$	20.7360	16.0467	41.5343	5.2572	32.0415	12.4487	23.6551
Gain	0.1028	0.1098	0.0830	0.0918	0.0866	0.0817	0.0850

4.9 Functional ANOVA: Case Study

In this section, we perform functional ANOVA analysis to Example 1.2 for the crank angle in $(360, 450)$. The model is FLM with kriging (4.20). Parameters θ and $\boldsymbol{\beta}$ are estimated by MLE from penalized likelihood followed by non-parametric smoothing along crank angle direction, which is described in previous sections. In order to see the effect of each variable, the functional ANOVA analysis is then performed to decompose the function onto each predictor orthogonally.

Figure 4.20 shows the estimated θ curve of MLE for penalized likelihood and the smoothed version, from where we can see that θ varies around 1.4. Note that the larger θ is, the weaker correlation between design cases is. This plot shows that the correlation around crank angle 360 and 400 is weaker than other crank angle region in $(360, 450)$. Figure 4.21 shows the corresponding estimated functional coefficients given the smoothed θ curve.

As before, we calculate the variance of linear effect terms and their ratio to the total variance to investigate how much the linear effect explains the total variance. Figure 4.22 is the total linear effect plot. It shows s versus t, where s is the sum of Sobol′ indices for linear effect: $s(t) = \sum_{i=1}^{d} s_i(t) = \frac{\sum_{i=1}^{d} D_i(t)}{D(t)}$. $s_i(t) = \frac{D_i(t)}{D(t)}$ is the Sobol′ indices for the ith linear effect. From this plot, we can see that the total linear effect explains over 85% of the overall variation at each point on the functional direction. Except around crank angle 395, this variance ratio is larger than 90% on

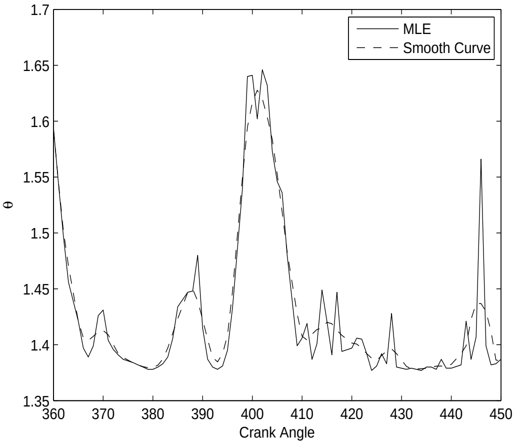
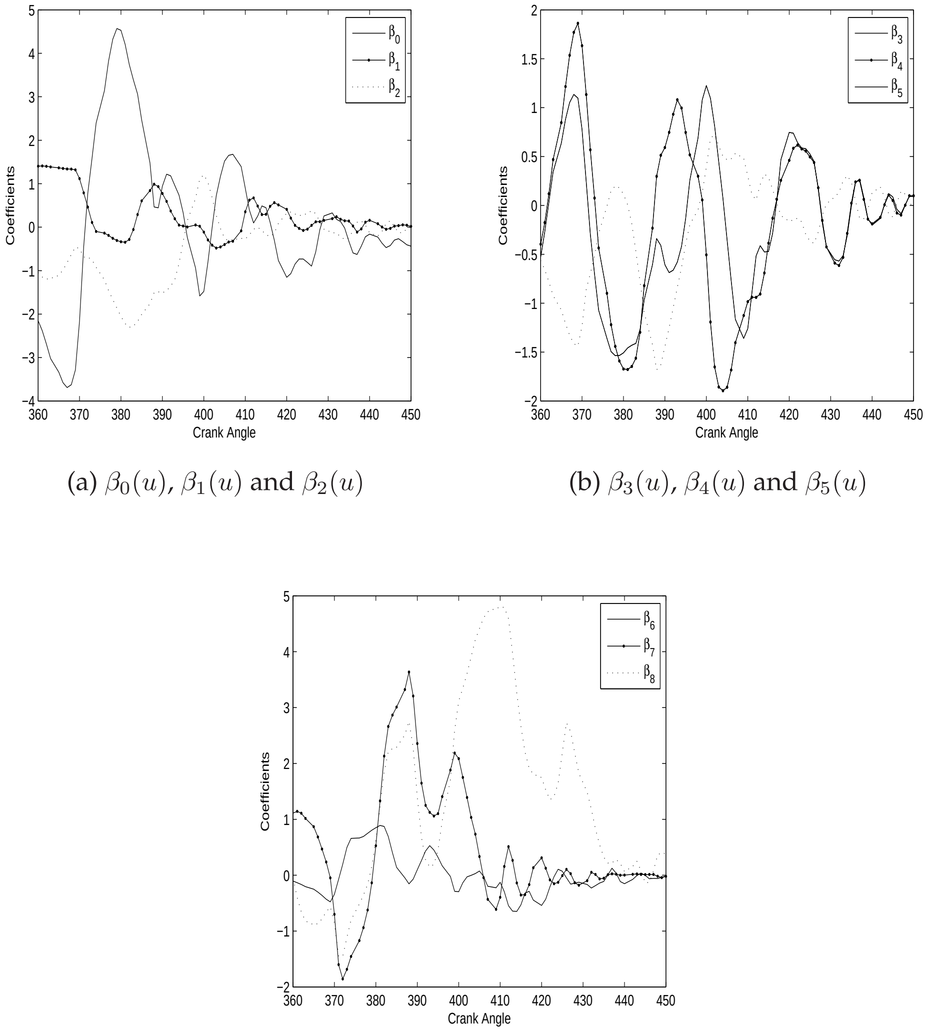


Figure 4.20. *MLE for penalized likelihood and the smoothed curve*

all other points. This plot shows that most of the overall variance is explained by the linear effects. Figure 4.23 shows each Sobol′ indices for individual linear component. We can see the linear effect on x_8 has the most impact on the overall variation. Its top variation range is around 0.8. The next one is x_7, which range is around 0.5. The next is x_2 and x_4, which range is around 0.35.

The above figures only give us information on how much each/total linear effect explain the total variance. To better understand how each variable affects the response, we need to get the estimation for each component function. After the functional ANOVA decomposition, each component g_i is a function of crank angle and the corresponding variable x_i which has been normalized and the range is in $(0, 1)$. We can calculate g_i at any given grid set on $(360, 450)$ by $(0, 1)$ to get an estimate of the component surface. Figures 4.24 and 4.25 show the contour plots for g_1 to g_8. We can see clearly how each component function varies on the predictor domain.



(a) $\beta_0(u)$, $\beta_1(u)$ and $\beta_2(u)$

(b) $\beta_3(u)$, $\beta_4(u)$ and $\beta_5(u)$

(c) $\beta_6(u)$, $\beta_7(u)$ and $\beta_8(u)$

Figure 4.21. *WLS estimates for the functional coefficients within crank angle region (360,450) in kriging model*

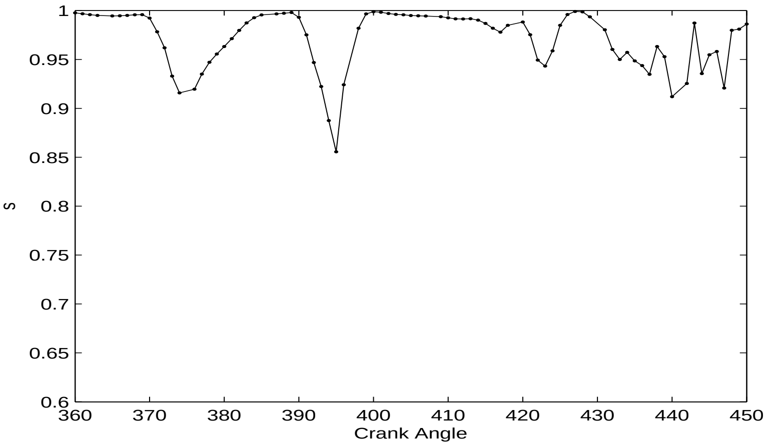
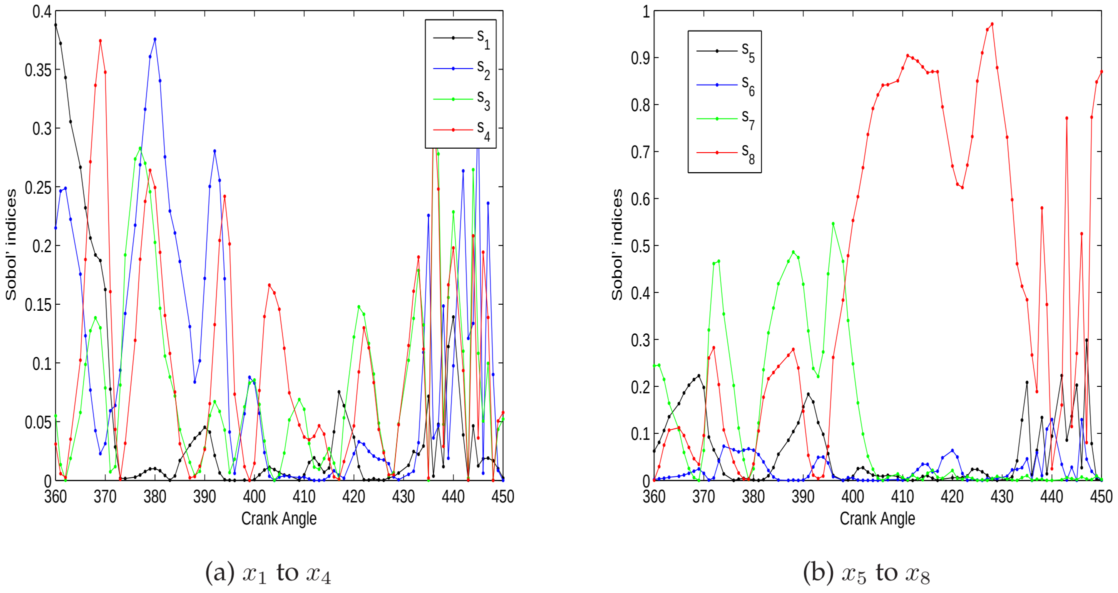


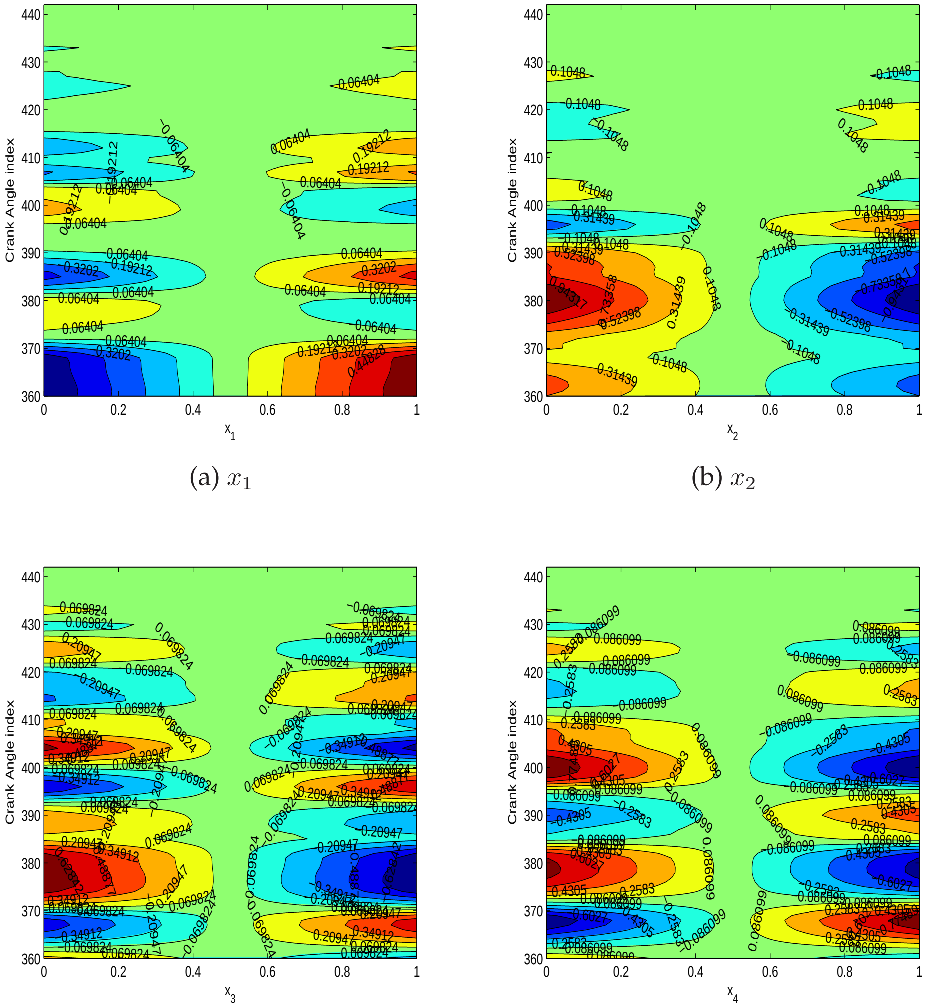
Figure 4.22. *Sum of Sobol′ indices for linear effect*



(a) x_1 to x_4

(b) x_5 to x_8

Figure 4.23. *Sobol′ indices for linear effect*



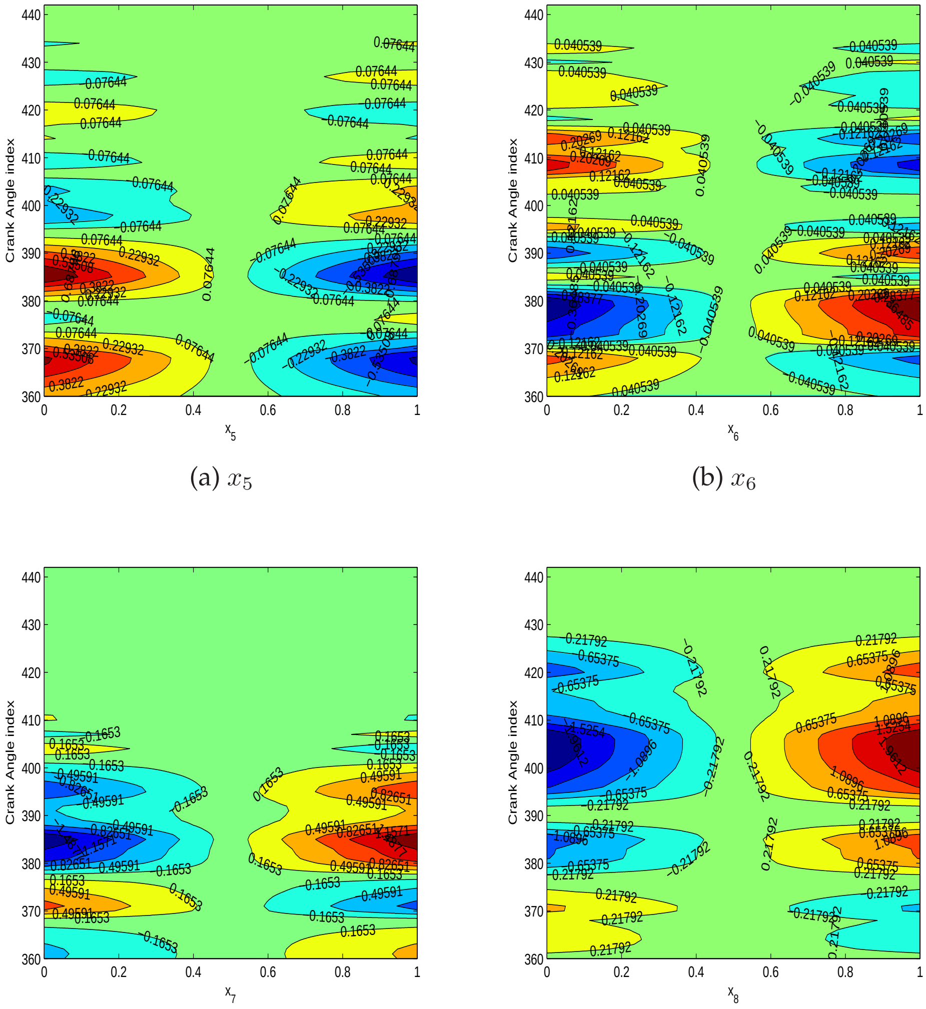
(a) x_1

(b) x_2

(c) x_3

(d) x_4

Figure 4.24. *Contour plots for x_1 to x_4*



(a) x_5

(b) x_6

(c) x_7

(d) x_8

Figure 4.25. *Contour plots for x_5 to x_8*

4.10 Conclusion

In this chapter, we propose a spatial-temporal model for functional response computer experiments with high sampling rate. This chapter focuses on the situation in which the overall trend can be represented as a functional linear model. We first model the overall trend using the functional linear model. We use two steps estimating procedure for FLM and the simulation shows the advantage of the smoothing step, which follows the step of initial least square estimation. Local linear regression with Epanechnikov kernel is implemented in smoothing step. The simulation shows the algorithm for FLM works quite well.

The proposed spatial-temporal model naturally produces a smooth interpolation surface over the spatial (design variable) and temporal (functional index) space, which is the most desirable functionality for engineers. We further compared two estimation and prediction procedures for this model, one is kriging with single response at each functional index followed by linear/cubic interpolation along functional index. This corresponds to the one proposed in Fang, Li and Sudjianto (2005). Another one is multivariate kriging over design variables and the nearest functional index points. Both procedures can predict at combination of any untried design variables and functional index, and interpolate the observed data points. The comparison result shows clearly that the multivariate version yields more accurate prediction than the single version.

We further developed the functional ANOVA decomposition for this spatial-temporal model, and derived a closed form for this decomposition. Since the sampling rate is intensive in the functional response situation, we can obtain a smoothed surface over the variable value range and functional direction for each variable from the functional ANOVA decomposition, which provides engineers more insights into how each variable affects the response in this situation.

We validated the proposed modeling procedures via a simulation study. Using the similar settings with valvetrain example, we generated several set of functional responses with different noise levels. The simulation results showed that the proposed estimation procedure performs quite well in the moderate noise level. Also, the variance of individual components is estimated with high accuracy in functional ANOVA decomposition.

For illustration purpose. we applied the proposed model and estimation procedure to valvetrain example for crank angle in $(360, 450)$. We recovered the coefficients of the main effects in spatial-temporal model and the parameters in correlation matrix in kriging terms. We found that the correlation around crank angle 360 and 400 is weaker than other crank angle region in $(360, 450)$. Furthermore, the total linear effect explains over 85% of the overall variation at each point on the functional direction. Except around crank angle 395, this variance ratio is larger than 90% on all other points. We also found that the linear effect on x_8 has the most impact on the overall variation. Its top variation range is around 0.8. The next one is x_7, which range is around 0.5. The next is x_2 and x_4, which range is around 0.35.

Finally, we obtained the contour plots for the functional response surfaces which is decomposed onto each variable. From those plots, we see clearly how each functional response component surface changes for each variable and functional index.

The methodology developed in this chapter can be applied directly to any computer experiments with output data with intensive sampling rate. The valvetrain example is a typical example of such computer experiments.

Chapter 5

Modeling Computer Experiment with Functional Response by Partial FLM

5.1 Partial Functional Linear Model

In most of the situations, we find that the functional coefficients have different levels of dependency on the functional index. Some of them vary a lot with it while others only change a little. As the same in previous chapter, denote n to be the number of observations for $(\mathbf{x}_i, y_i)$ and let n_i denote the number of functional points at each observation point i. If in a functional linear model defined in (4.3), some of the coefficients in $\boldsymbol{\beta}$ strongly depend on the functional index (denote it by u) compared to the rest of the coefficients in $\boldsymbol{\beta}$, then we can split $\boldsymbol{\beta}$ into two groups, one group includes all the coefficients which are highly dependent on u, the other group contains the rest of the coefficients. Let $\boldsymbol{\alpha}$ denote the coefficients in the first group and **w** denote the corresponding predictors. Let $\boldsymbol{\gamma}$ denote the rest of the coefficients and **z** denote the corresponding predictors. For the predictors **z**, we can simplify their coefficients as constants so that the functional linear model turns into:

$$\mu(\mathbf{x}, u) = \mathbf{w}\boldsymbol{\alpha}(u) + \mathbf{z}\boldsymbol{\gamma}, \tag{5.1}$$

where $\mathbf{x} = (\mathbf{w}, \mathbf{z})$. In this model, part of the coefficients $(\boldsymbol{\alpha}(u))$ are functions of u, the other part $\boldsymbol{\gamma}$ are just constants. So this model are called semi-varying coefficient model.

With the mean function (5.1), model (4.3) becomes:

$$y(\mathbf{x}, u) = \mathbf{w}\boldsymbol{\alpha}(u) + \mathbf{z}\boldsymbol{\gamma} + \epsilon(\mathbf{x}, u) \tag{5.2}$$

We can write this model at observation point $(\mathbf{x}_i, y_i)$ at time u_j componentwise as:

$$y_i(u_j) = \mathbf{w}_i\boldsymbol{\alpha}(u_j) + \mathbf{z}_i\boldsymbol{\gamma} + \epsilon_{i,j} \tag{5.3}$$

5.2 Back-fitting Algorithm

The idea of back-fitting algorithm is very simple. If we know $\boldsymbol{\gamma}$, then using the procedure for the functional linear model, we can solve for $\boldsymbol{\alpha}(u)$ for the following model:

$$\begin{aligned} y^f(\mathbf{x}, u) &= y(\mathbf{x}, u) - \mathbf{z}\boldsymbol{\gamma} \\ &= \mathbf{w}\boldsymbol{\alpha}(u) + \epsilon(\mathbf{x}, u) \end{aligned} \tag{5.4}$$

And if we know $\boldsymbol{\alpha}(u)$, then we can use least square estimator to solve for $\boldsymbol{\gamma}$ for the following model:

$$y^l(\mathbf{x}, u) = y(\mathbf{x}, u) - \mathbf{w}\boldsymbol{\alpha}(u) = \mathbf{z}\boldsymbol{\gamma} + \epsilon(\mathbf{x}, u) \tag{5.5}$$

The above steps immediately give us an iterative method to solve for $\boldsymbol{\alpha}(u)$ and $\boldsymbol{\gamma}$ simultaneously. To get the initial value for the iteration procedure, we can get the least square estimators for the coefficients $(\boldsymbol{\alpha}(u_j)^T, \boldsymbol{\gamma}(u_j)^T)^T$ at each time point u_j, denote them as $\hat{\boldsymbol{\alpha}}^{(0)}(u_j)$ and $\hat{\boldsymbol{\gamma}}^{(0)}(u_j)$:

$$\begin{pmatrix} \hat{\boldsymbol{\alpha}}^{(0)}(u_j) \\ \hat{\boldsymbol{\gamma}}^{(0)}(u_j) \end{pmatrix} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}(u_j)$$

where $\mathbf{X} = (\mathbf{w}^T, \mathbf{z}^T)^T$ is the design matrix for the whole system, $\mathbf{Y}(u_j) = (y_1(u_j), \cdots, y_n(u_j))^T$ is the observed response vector at time u_j.

For the back-fitting algorithm, first calculate:

$$y_i^l(u_j) = y_i(u_j) - \mathbf{w}_i\hat{\boldsymbol{\alpha}}^{(0)}(u_j)$$

and average $y_i^l(u_j)$ along time direction:

$$y_i^l = \frac{1}{n_i}\sum_{j=1}^{n_i} y_i^l(u_j)$$

then using least square to get $\hat{\boldsymbol{\gamma}}^{(1)}$ from the following model:

$$\mathbf{Y}^l = \mathbf{Z}\boldsymbol{\gamma} + \varepsilon$$

where **Z** is the design matrix composed only by the predictor group **z**, $\mathbf{Y}^l = (y_1^l, y_2^l, \cdots, y_n^l)^T$ and:

$$\hat{\boldsymbol{\gamma}}^{(1)} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{Y}^l$$

The second step is to get $\hat{\boldsymbol{\alpha}}^{(0)}(u_j)$ for all the time point. First calculate:

$$y_i^f(u_j) = y_i(u_j) - \mathbf{z}_i\hat{\boldsymbol{\gamma}}^{(1)}$$

then using functional linear model to estimate the coefficient $\boldsymbol{\alpha}(u_j)$ from the following model:

$$y_i^f(u_j) = \mathbf{w}_i\boldsymbol{\alpha}(u_j) + \epsilon_{i,j}$$

and denote the resulting estimates for $\boldsymbol{\alpha}(u_j)$ as $\hat{\boldsymbol{\alpha}}^{(1)}(u_j)$.

Theses two steps complete an iteration step. Continue to iterate for these two steps until the solution converges. Usually this iterative method converges very fast.

5.3 Simulations for Back-fitting Algorithm

As an illustrating example, we generate a random sample of size $n = 16$ and $n_i = 245$ from the following semi-varying coefficient model:

$$y_i(u_j) = \alpha_0(u_j) + w_{1,i}\alpha_1(u_j) + w_{2,i}\alpha_2(u_j) + \sum_{k=1}^{7} z_{k,i}\gamma_k + \epsilon_{i,j}, \tag{5.6}$$

where $\alpha_0(u) = \sin(2\pi u)$, $\alpha_1(u) = \cos(2\pi u)$, $\alpha_2(u) = 4(u-0.5)^2$, $u_j = \frac{j}{n_i+1}$, $w_{1,i}, w_{2,i}, z_{k,i} \sim Bernoulli(0.5)$, $\gamma_k = k/8$, $k = 1, \cdots, 7$ and $\epsilon_{i,j} \sim N(0, \sigma^2)$. Note that we purposefully

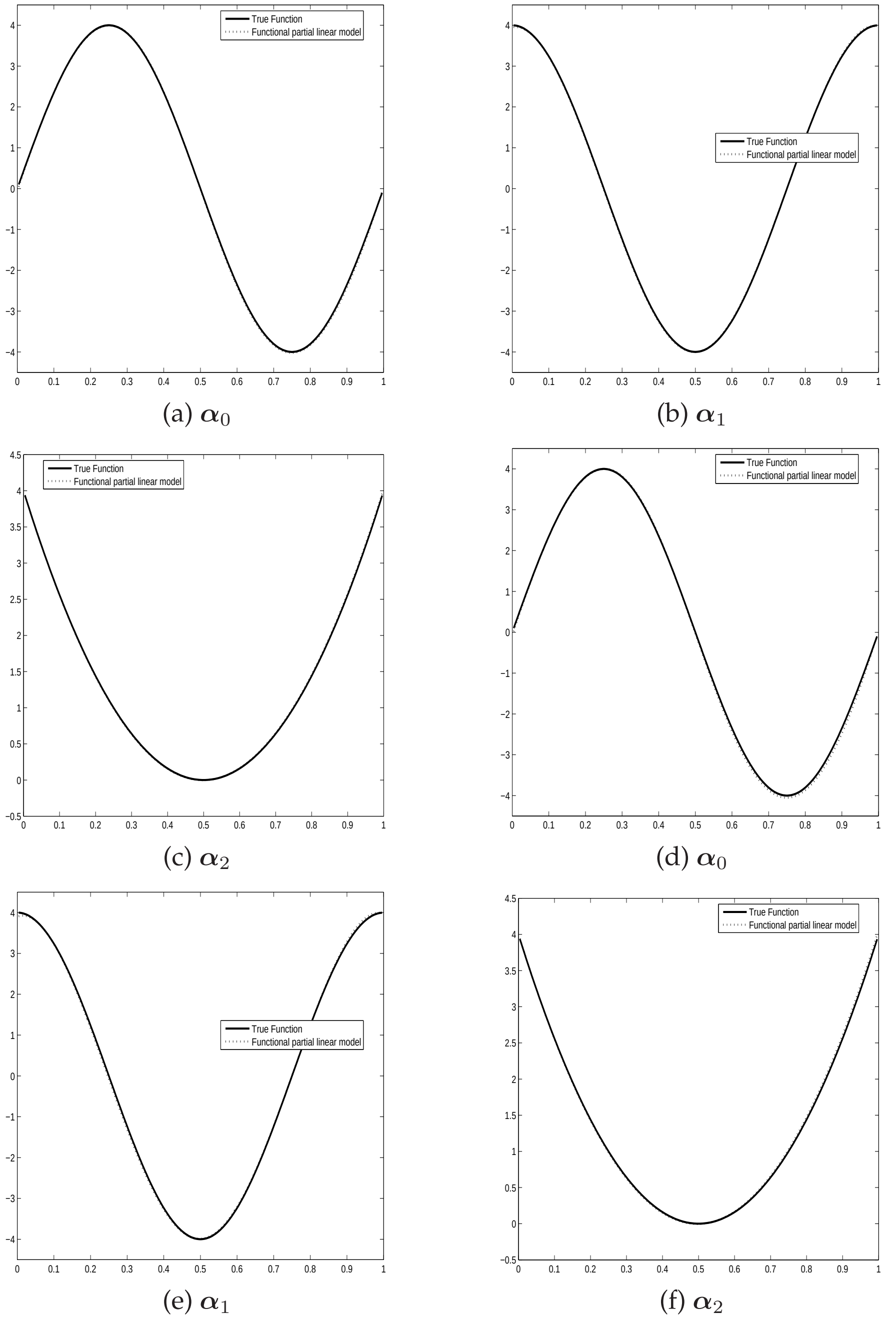
Table 5.1. *γ and $\hat{\gamma}$ from semi-varying coefficient model*

k	γ_k	$\sigma^2 = 0.25$	$\sigma^2 = 1$	$\sigma^2 = 2$	$\sigma^2 = 4$
$k = 1$	0.1250	0.1326	0.1402	0.1465	0.1554
$k = 2$	0.2500	0.2614	0.2729	0.2824	0.2958
$k = 3$	0.3750	0.3692	0.3634	0.3586	0.3518
$k = 4$	0.5000	0.5020	0.5039	0.5056	0.5079
$k = 5$	0.6250	0.6219	0.6188	0.6162	0.6125
$k = 6$	0.7500	0.7555	0.7609	0.7655	0.7719
$k = 7$	0.8750	0.8843	0.8935	0.9012	0.9120

matched the number of observations and data points in functional direction as the case in Example 1.2 to study the algorithm quality.

As in the previous chapters, we tested several cases with different variance values $\sigma^2 = 0.25, 1, 2$ and 4. Table 5.1 lists the "true" constant coefficients of γ_k, $k = 1, \cdots, 7$ and the converged solution for the estimated value $\hat{\gamma}_k$. From this table, we can see the estimated result is very close to the true value even for the large variance when $\sigma^2 = 4$.

Figure 5.1 shows the fitted functional coefficient for $\boldsymbol{\alpha}_0$, $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ for $\sigma^2 = 0.25$ and 1. We can see that the estimates and the original true mean function almost collapse into one curve. Figure 5.2 shows the fitted functional coefficient for $\boldsymbol{\alpha}_0$, $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ for $\sigma^2 = 2$ and 4. We can see even for the large variance in error term like $\sigma^2 = 4$, the estimates and the original true mean function are very close. This simulation result proves the quality of the backfitting algorithm for partial functional linear model.



(a) $\boldsymbol{\alpha}_0$

(b) $\boldsymbol{\alpha}_1$

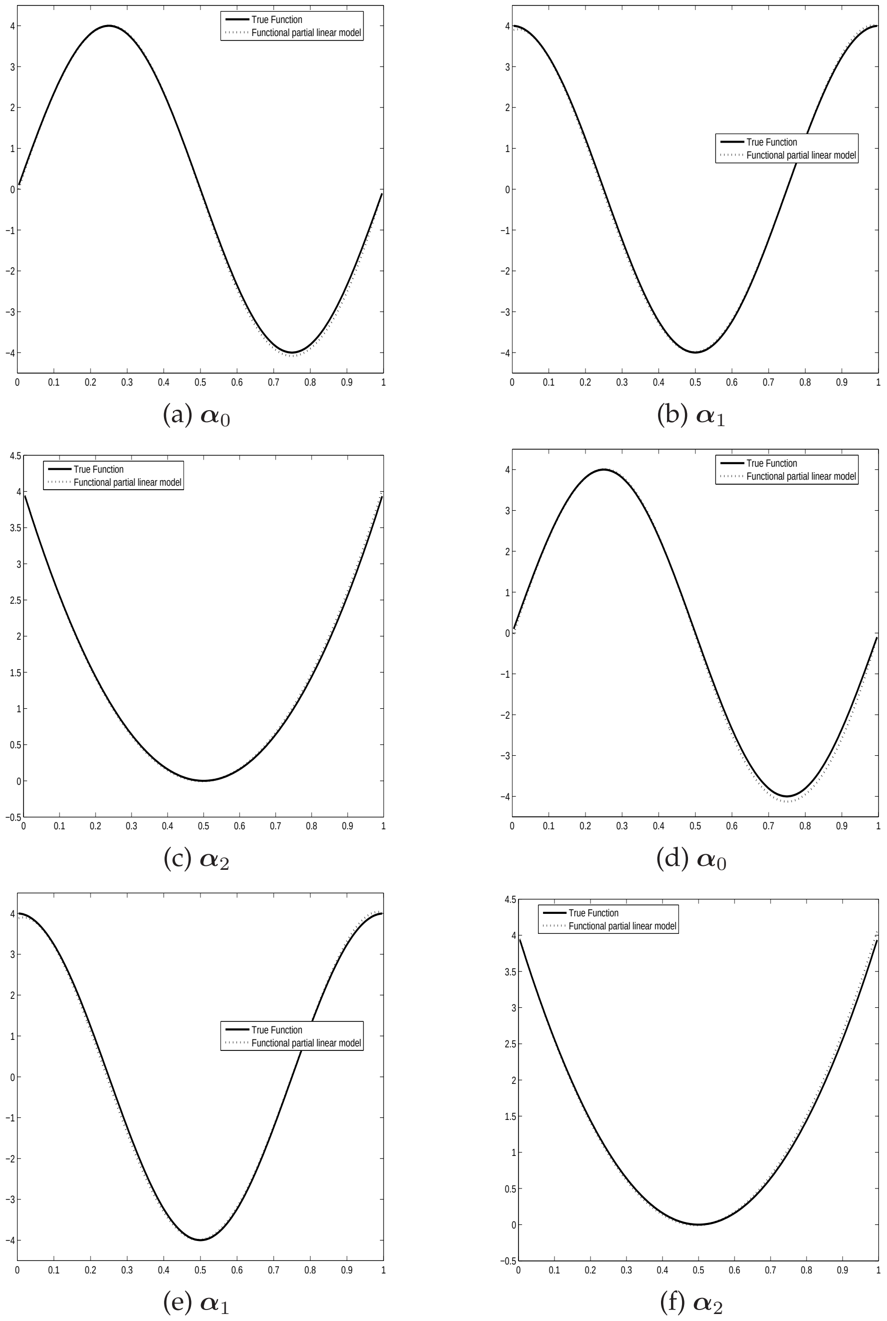
(c) $\boldsymbol{\alpha}_2$

(d) $\boldsymbol{\alpha}_0$

(e) $\boldsymbol{\alpha}_1$

(f) $\boldsymbol{\alpha}_2$

Figure 5.1. *Estimates of functional coefficient for model (5.6) with σ^2=0.25 and 1. (a), (b) and (c) are for σ^2=0.25; (d), (e) and (f) are for σ^2=1*



(a) $\boldsymbol{\alpha}_0$

(b) $\boldsymbol{\alpha}_1$

(c) $\boldsymbol{\alpha}_2$

(d) $\boldsymbol{\alpha}_0$

(e) $\boldsymbol{\alpha}_1$

(f) $\boldsymbol{\alpha}_2$

Figure 5.2. *Estimates of functional coefficient for model (5.6) with σ^2=2 and 4. (a), (b) and (c) are for σ^2=2; (d), (e) and (f) are for σ^2=4*

5.4 Partial Functional Linear Model with Kriging and Functional ANOVA Decomposition

The previous sections concentrate on the mean function estimation for partial functional linear model, which is a much more parsimonious model than FLM. Since it is a regression model, the mean function will not interpolate the observed value. But in engineering field, people always want to get a model which can interpolate the data.

The kriging model for partial FLM is a natural extension of model (4.20). Simply replace the mean function $\mathbf{x}\boldsymbol{\beta}(u)$ in model (4.20) with the mean function of partial FLM: $\mathbf{w}\boldsymbol{\alpha}(u) + \mathbf{z}\boldsymbol{\gamma}$, we have the partial FLM with kriging:

$$m(\mathbf{x}; u) = \mathbf{w}\boldsymbol{\alpha}(u) + \mathbf{z}\boldsymbol{\gamma} + \mathbf{r}(\mathbf{x}; u)\mathbf{R}^{-1}(u)(\mathbf{Y}(u) - \mathbf{w}\boldsymbol{\alpha}(u) - \mathbf{z}\boldsymbol{\gamma}). \tag{5.7}$$

We propose the estimation procedure for model (5.7) as follows. First, treat the model (5.7) as a general FLM:

$$\mathbf{y}(u) = \mathbf{x}\boldsymbol{\beta}(u) + \epsilon(u) \sim N(\mathbf{x}\boldsymbol{\beta}(u), \sigma^2(u)\mathbf{R}(u)), \tag{5.8}$$

where $\mathbf{x} = (\mathbf{w}, \mathbf{z})$. Using the estimation procedure in Chapter 4, we can get the MLE for all the parameters in this model , such as the functional coefficients $\boldsymbol{\beta}(u)$, the parameters $\theta(u)$ in correlation matrix $\mathbf{R}(u)$, and the functional variance $\sigma^2(u)$.

Next, we rewrite $\mathbf{x} = (\mathbf{w}, \mathbf{z})$, split the coefficients into functional group $\boldsymbol{\alpha}(u)$ and constant group $\boldsymbol{\gamma}$, and substitute the estimated variance $\hat{\sigma}^2(u)$ and correlation matrix $\hat{\mathbf{R}}(u)$ into the above model, then the FLM changes into partial FLM with known variance-covariance structure:

$$\mathbf{y}(u) = \mathbf{w}\boldsymbol{\alpha}(u) + \mathbf{z}\boldsymbol{\gamma} + \epsilon(u) \sim N(\mathbf{w}\boldsymbol{\alpha}(u) + \mathbf{z}\boldsymbol{\gamma}, \hat{\sigma}^2(u)\hat{\mathbf{R}}(u)). \tag{5.9}$$

Next, we invoke backfitting algorithm to get the estimation for the coefficients in model (5.9):

- set $\boldsymbol{\alpha}(u) = \hat{\boldsymbol{\alpha}}(u)$, $\mathbf{Y}^*(u) = \mathbf{Y}(u) - \mathbf{w}\hat{\boldsymbol{\alpha}}(u) = \mathbf{z}\boldsymbol{\gamma} + \epsilon(u)$. By assuming the independent of $\epsilon(i)$ and $\epsilon(j)$ for $i \neq j$, the estimate of $\boldsymbol{\gamma}$ is:

$$\hat{\gamma} = (\mathbf{z}' \sum_{u=1}^{J} \text{Cov}\epsilon(u)^{-1}\mathbf{z})^{-1}(\mathbf{z}' \sum_{u=1}^{J} \text{Cov}\epsilon(u)^{-1}\mathbf{Y}^*(u)), \tag{5.10}$$

where $\text{Cov}\epsilon(u) = \sigma^2(u)\mathbf{R}(u)$. In the estimation procedure, we substitute the estimated variance and correlation matrix into this formula.

- set $\boldsymbol{\gamma} = \hat{\gamma}$, $\mathbf{Y}^{**}(u) = \mathbf{Y}(u) - \mathbf{z}\hat{\gamma} = \mathbf{w}\boldsymbol{\alpha}(u) + \epsilon(u)$, the WLE for $\boldsymbol{\alpha}$ is:

$$\hat{\boldsymbol{\alpha}}(u) = (\mathbf{w}'\mathbf{R}^{-1}(u)\mathbf{w})^{-1}\mathbf{w}'\mathbf{R}^{-1}(u)\mathbf{Y}^{**}(u). \tag{5.11}$$

We get the final estimation result by repeating above two steps until the algorithm converges.

Functional ANOVA decomposition for partial FLM with kriging is also a natural extension of the functional ANOVA decomposition in FLM case. After having all the estimates for functional coefficients $\boldsymbol{\alpha}(u)$, the constant coefficients $\boldsymbol{\gamma}$, and parameters $\boldsymbol{\theta}$ in correlation matrix $\mathbf{R}$, the constant coefficient $\boldsymbol{\gamma}$ can be viewed as a function of u with constant value so that we can substitute these estimates into formula (4.29) to (4.32) for FLM with kriging case.

5.5 Simulation for Partial FLM with Kriging and Functional ANOVA Decomposition

In order to test the quality of the code and the algorithm for partial FLM with kriging, we generate simulation data by mimicking Example 1.2:

$$y_{ij} = \alpha_{0,j} + w_{i,1}\alpha_{1,j} + w_{i,2}\alpha_{2,j} + \sum_{k=1}^{6} z_{i,k}\gamma_k + \epsilon_{ij}, \tag{5.12}$$

where $i = 1, \cdots, 16$, $j = 1, \cdots, 253$, $w_{:,1} = \mathbf{X}_8$, $w_{:,2} = \mathbf{X}_9$, $z_{:,k} = \mathbf{X}_{k+1}$, and $\mathbf{X}_k$ is taken as the design case in Example 1.2, which is given in table 4.1. $\gamma_k = 1 + \sin(\frac{k}{7})$ for $k = 1, \cdots, 6$. The functional coefficients $\alpha_{j,:}$ are given in figure (5.3), where $j = 0, 1, 2$.

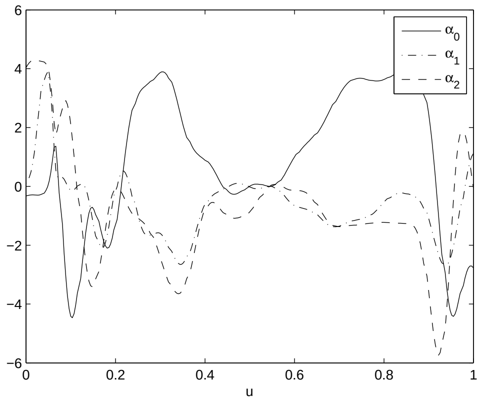


Figure 5.3. *Functional coefficients* α_0,α_1,α_2 *in simulation 5.12*

The error term needs to be smooth, which is required by the kriging model. We use the same error term as in the simulation for FLM:

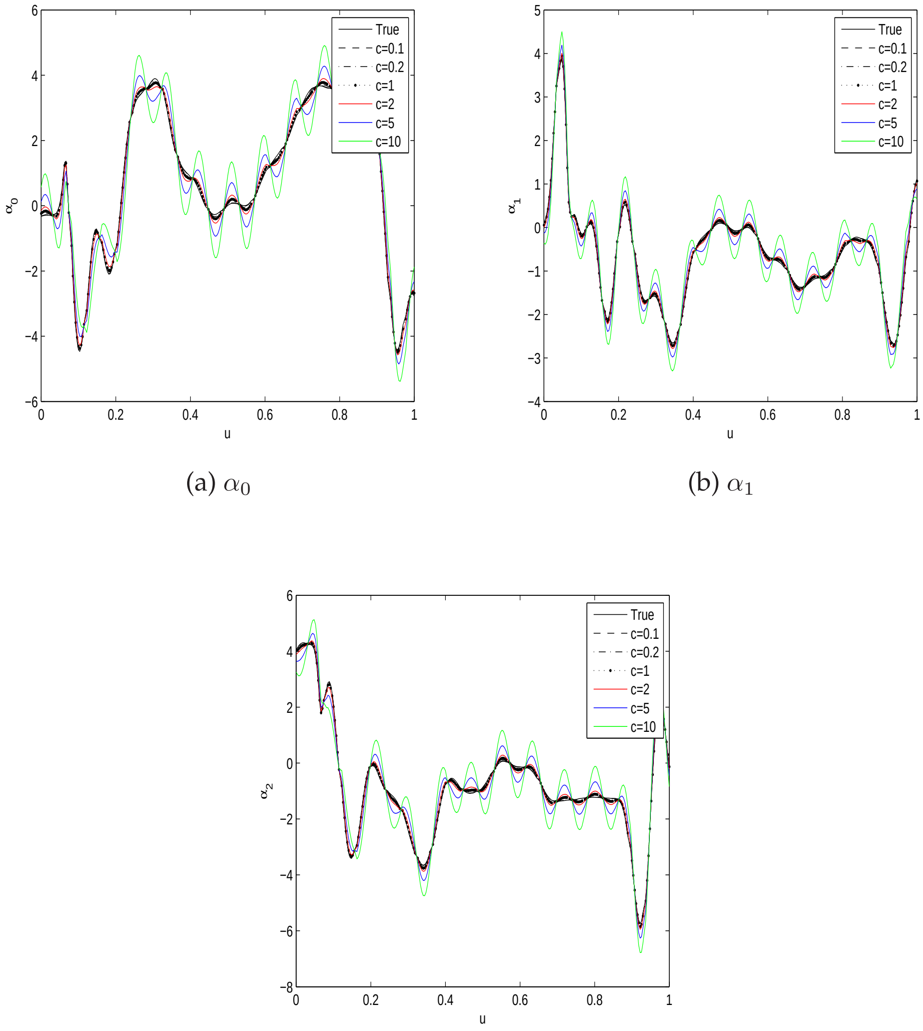
$$\epsilon_{ij} = c[\sin(2k\pi u_j) + \cos(2k\pi u_j)]exp(-\frac{1}{2}\mathbf{x}_i\mathbf{A}_j\mathbf{x}_i'), \tag{5.13}$$

where $i = 1, 2, \cdots, 16$, $j = 1, \cdots, 253$. $\mathbf{x}_i$ is the design variable corresponding to the ith design case, which is listed in table 4.1. $\mathbf{A}_j$ is a square matrix with off diagonal elements equal to ρ_j, and diagonal element equal to unit. ρ_j is uniformly randomly picked within range $(0.3, 0.7)$. $\mathbf{A}_j$ can be viewed as the correlation matrix for $\mathbf{x}_i$ at the $j-th$ functional index. As the same with FLM case, we take $k = 12$, and error level c as: 0.01,0.1,0.5,1,2,5,10.

First of all, the estimated functional coefficients are compared with the true values. Table 5.2 lists the MSE of estimated functional coefficients. We can see that even for the extreme case where noise amplitude $c = 10$, MSE is less than unity. Figure 5.4 gives the simulation result for α_0, α_1 and α_2 under several noise levels. We can see that the estimated functional coefficients is very close to the true curve even for the largest noise level.

Table 5.2. *MSE for functional coefficients in simulation 5.12*

MSE	$c = 0.01$	$c = 0.1$	$c = 0.2$	$c = 1$	$c = 2$	$c = 5$	$c = 10$
α_0	0.0009	0.0094	0.0472	0.0946	0.1889	0.4735	0.9444
α_1	0.0005	0.0047	0.0233	0.0467	0.0934	0.2340	0.4669
α_2	0.0008	0.0080	0.0402	0.0804	0.1608	0.4026	0.8038



(a) α_0

(b) α_1

(c) α_2

Figure 5.4. *Simulation result for functional coefficients for simulation 5.12*

Table 5.3. *Error for estimated constant coefficients in simulation 5.12*

	True	$c = 0.01$	$c = 0.1$	$c = 0.2$	$c = 1$	$c = 2$	$c = 5$	$c = 10$
γ_1	1.1424	0.1148E-04	0.1143E-03	0.0006	0.0011	0.0023	0.0057	0.0114
γ_2	1.2818	0.1148E-04	0.1143E-03	0.0006	0.0011	0.0023	0.0057	0.0114
γ_3	1.4156	0.1008E-04	0.1005E-03	0.0005	0.0010	0.0020	0.0050	0.0100
γ_4	1.5408	0.1008E-04	0.1005E-03	0.0005	0.0010	0.0020	0.0050	0.0100
γ_5	1.6551	0.1008E-04	0.1005E-03	0.0005	0.0010	0.0020	0.0050	0.0100
γ_6	1.7560	0.1008E-04	0.1005E-03	0.0005	0.0010	0.0020	0.0050	0.0100

Next, we consider the quality for estimating the constant coefficients in simulation 5.12. The error for this estimation is calculated. The error is defined by $\hat{\gamma}_i - \gamma_i$ for $i = 1, \cdots, 6$. Table 5.3 lists this error information for the simulation under different noise level. We can see that even for the largest noise level $c = 10$, the estimated constant coefficients are very close to the true value.

Next, we consider the quality of the functional ANOVA decomposition. Since we know the true function of (5.12), we can integrate it and get the true decomposed functional components and their variances. Then we can calculate the MSE for each estimated decomposed functional component along functional direction. Table 5.4 gives the MSE for estimated variance of individual decomposed functional components along functional direction. This MSE is defined as:

$$MSE_i = \sqrt{\frac{1}{J}\sum_{j=1}^{J}(\hat{\text{Var}}g_{i,j} - \text{Var}g_{i,j})^2},$$

where $g_{i,j}$ is the ith decomposed functional component at the jth functional index, Var$g_{i,j}$ is the variance corresponding to this function, $\hat{\text{Var}}g_{i,j}$ is the estimated variance value. Figure 5.5 gives the true and estimated total variance for the functional ANOVA decomposing. Together with table 5.4, we can see that when noise/signal level is not too extreme, the estimation is good. Figures 5.6 and 5.7 show the estimated

Table 5.4. *MSE for individual decomposed functional component in simulation 5.12*

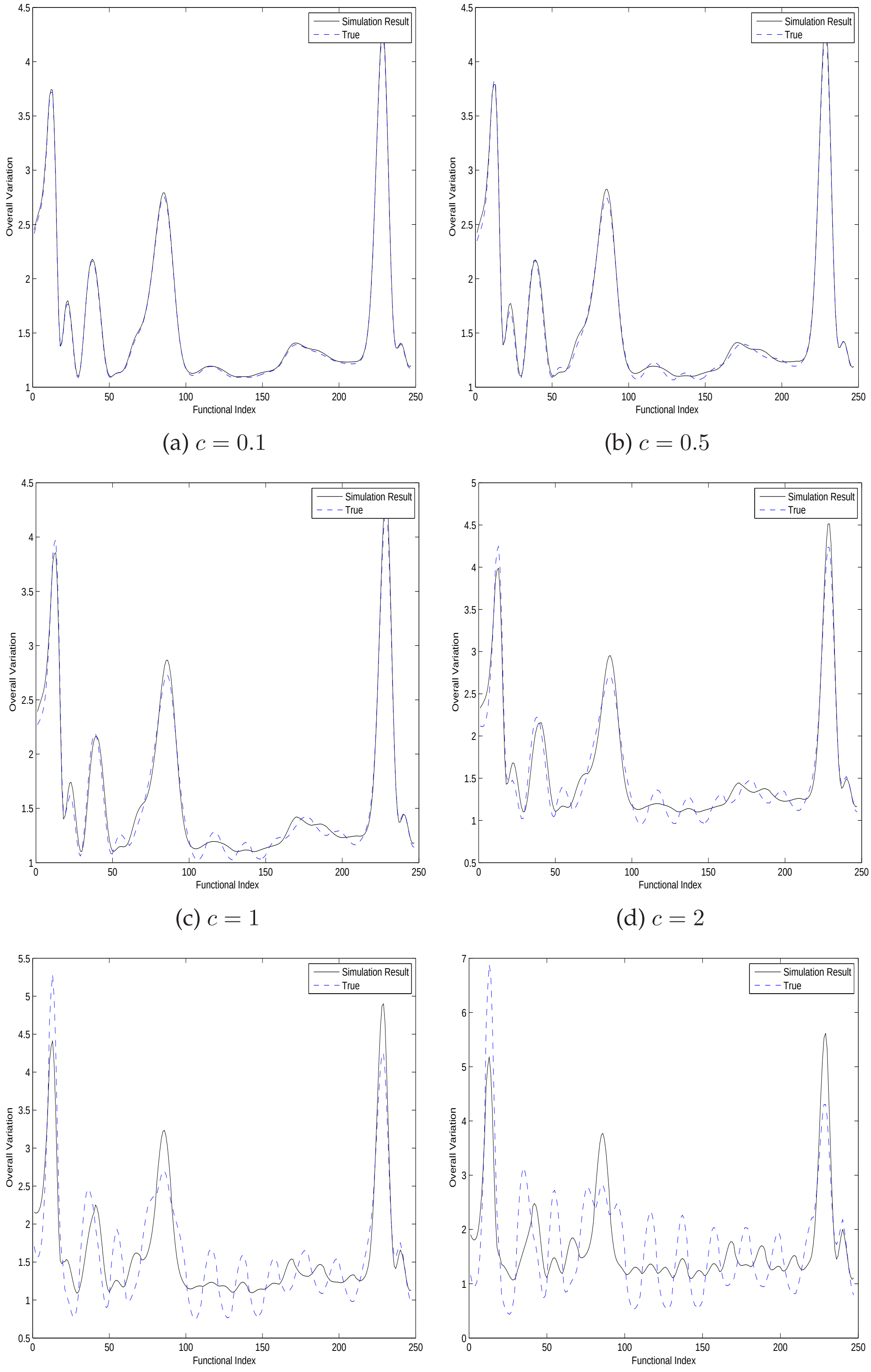
MSE	$c = 0.01$	$c = 0.1$	$c = 0.2$	$c = 1$	$c = 2$	$c = 5$	$c = 10$
g_1	0.0025	0.0023	0.0023	0.0038	0.0070	0.0217	0.0406
g_2	0.0066	0.0071	0.0106	0.0155	0.0277	0.0648	0.1376
g_3	0.0011	0.0013	0.0035	0.0071	0.0129	0.0333	0.0640
g_4	0.0013	0.0016	0.0039	0.0080	0.0145	0.0372	0.0709
g_5	0.0016	0.0019	0.0047	0.0095	0.0172	0.0440	0.0839
g_6	0.0019	0.0022	0.0052	0.0104	0.0188	0.0479	0.0907
g_7	0.0022	0.0025	0.0056	0.0112	0.0202	0.0514	0.0972
g_8	0.0025	0.0028	0.0060	0.0119	0.0214	0.0545	0.1029
Total Variance	7.1407E-04	0.0071	0.0357	0.0745	0.1424	0.3719	0.7677

variance for decomposed $g_1(w_1)$ and $g_2(w_2)$, from which we can see the estimation for each component variance is good.

Since there is no main effect for variables z_j ($j = 1, \cdots, 6$), the decomposed variance comes from the error term $\epsilon_{i,j}$ in (5.12). The noise to signal ratio here is very large even for small value of c. Hence the variance change in functional direction will be influenced a lot by error. Figure 5.8 gives the comparison of the estimated and true variance of decomposed $g_3(z_1)$.

5.6 Simulation–Comparison of Multivariate and Single Response

In this section, we make a comparison between multivariate kriging model and kriging model with single response. Equation (5.12) is used to generate the simulation response on the given design matrix in table 4.1. Furthermore, we generate another N random observations for the same predictors, where $N = 1000$. Using PFLM, we obtain the main functions estimates and then get the residual. We label these points as $\mathbf{x}_i^*$, where $i = 1, \cdots, N$, and the corresponding functional residual



(a) $c = 0.1$

(b) $c = 0.5$

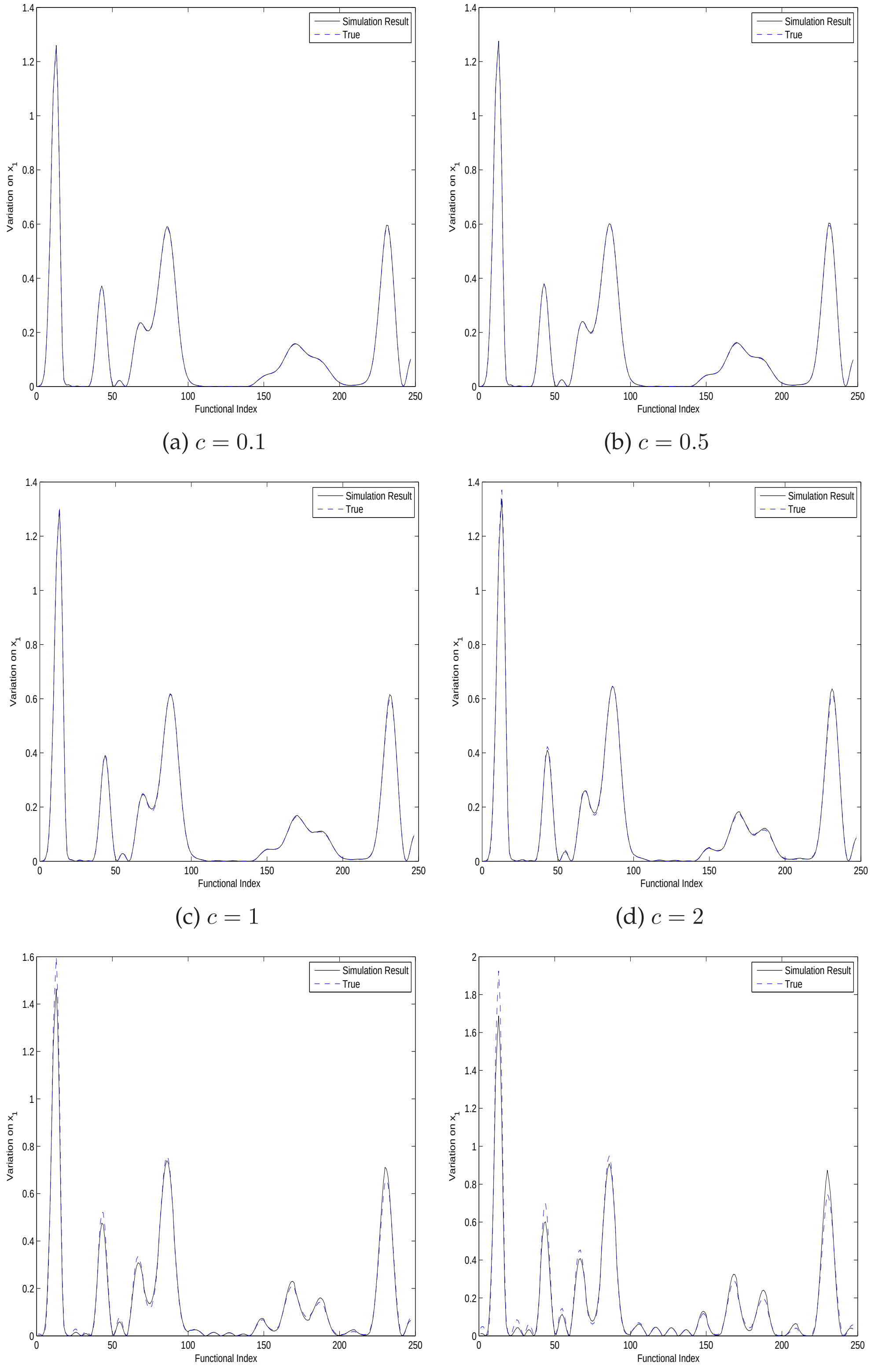
(c) $c = 1$

(d) $c = 2$

(e) $c = 5$

(f) $c = 10$

Figure 5.5. *Estimated total variance for functional ANOVA decomposition for simulation 5.12*



(a) $c = 0.1$

(b) $c = 0.5$

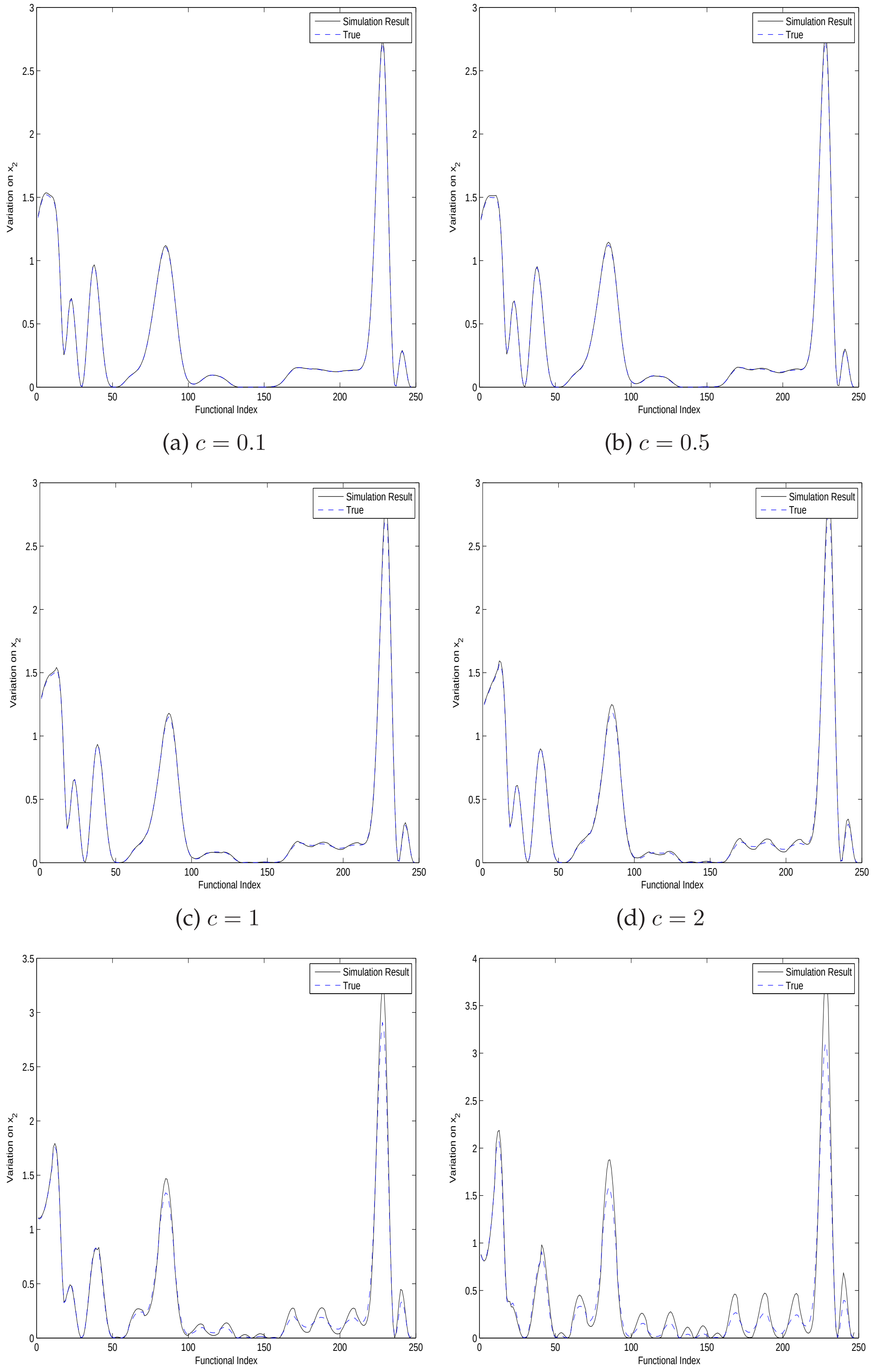
(c) $c = 1$

(d) $c = 2$

(e) $c = 5$

(f) $c = 10$

Figure 5.6. *Estimated variance for decomposed $g_1(w_1)$ for simulation 5.12*



(a) $c = 0.1$

(b) $c = 0.5$

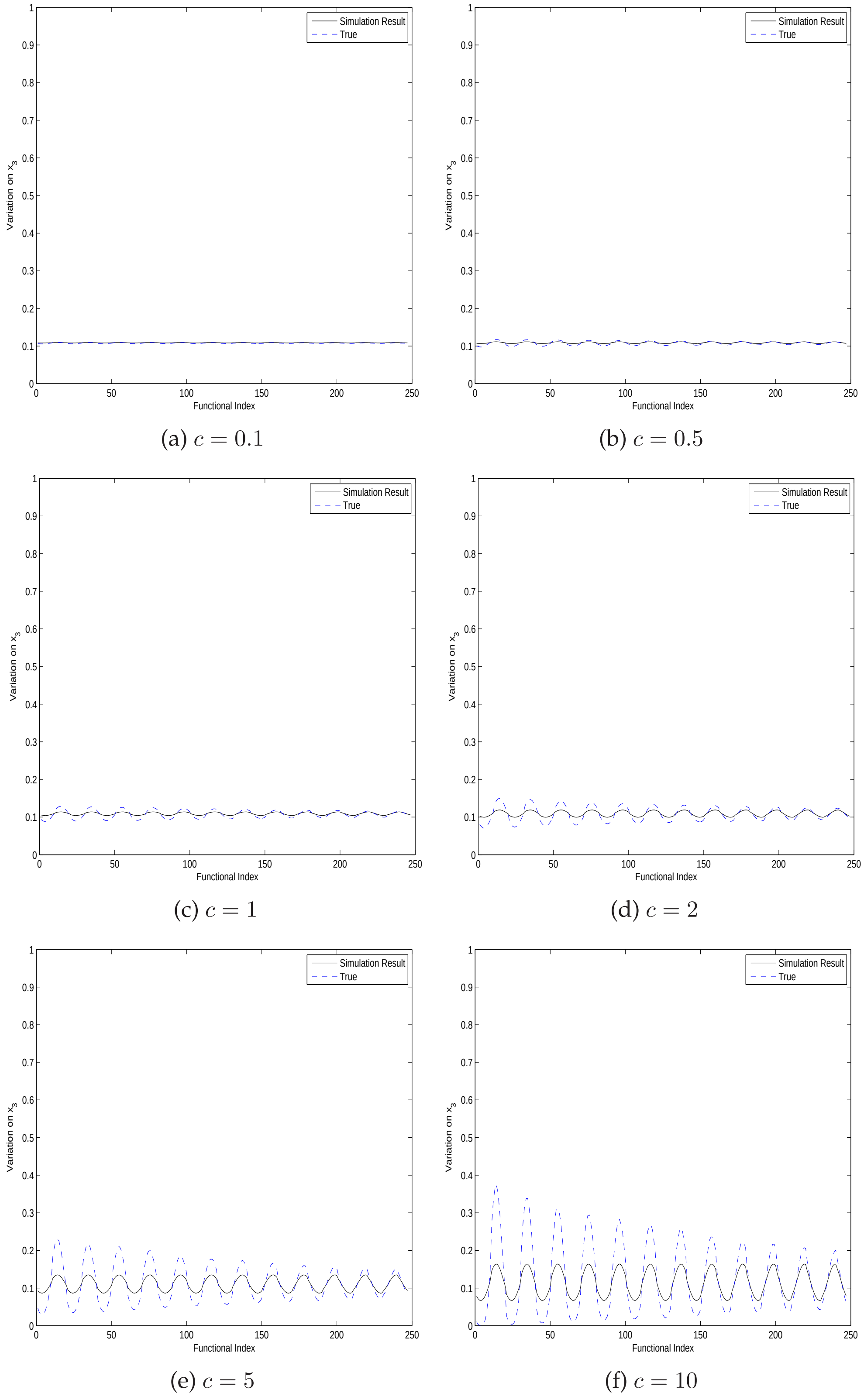
(c) $c = 1$

(d) $c = 2$

(e) $c = 5$

(f) $c = 10$

Figure 5.7. *Estimated variance for decomposed $g_2(w_2)$ for simulation 5.12*



(a) $c = 0.1$

(b) $c = 0.5$

(c) $c = 1$

(d) $c = 2$

(e) $c = 5$

(f) $c = 10$

Figure 5.8. *Estimated variance for decomposed $g_3(z_1)$ for simulation 5.12*

Table 5.5. *Comparison for multivariate kriging and kriging with single response in PFLM for* $c = 10$

$Source$	$j = 1$	$j = 3$	$j = 5$	$j = 7$	$j = 9$	$j = 11$	$j = 13$
$P.E.(single)$	17.0364	22.1657	15.5470	43.5936	23.6791	11.5649	16.1706
$P.E.(multivariate)$	14.9903	19.5102	13.4279	35.3501	20.0254	9.8105	13.5073
Gain	0.1201	0.1198	0.1363	0.1891	0.1543	0.1517	0.1647

as $\mathbf{y}_i^*$. Then we use kriging with single response and multivariate kriging model to build models on design matrix separately. Then the predictions on each of the randomly generated points are obtained from each model. The prediction errors for each functional index are defined as

$$P.E.(j) = \sum_{i=1}^{N} \|(y^*(j)_i - \hat{y}^*(j)_i)\|^2,$$

where $y^*(j)$ is the ith residual at functional index j, $\hat{y}^*(j)$ is the predicted value.

We use 2 points in functional direction for multivariate version to compare with the single response version. The result shows that the prediction error of single version is uniformly greater than the one of multivariate version. The averaged prediction error across functional direction for single version is 18.6301 for $c = 10$ (the magnitude for the error term), while the corresponding result for multivariate version is 16.7326. Table 5.5 shows the detailed comparison for some functional index j, where gain is defined as $\frac{P.E.(single) - P.E.(multivariate)}{P.E.(single)}$. From this table, we can see we gain a lot by using multivariate kriging model compared to kriging with single response. Also note that for different functional index, we have difference gains. This clearly shows the advantage of multivariate kriging model.

5.7 Case Study: Partial Functional Linear Model with kriging for valve motion error within crank angle $(90, 360)$

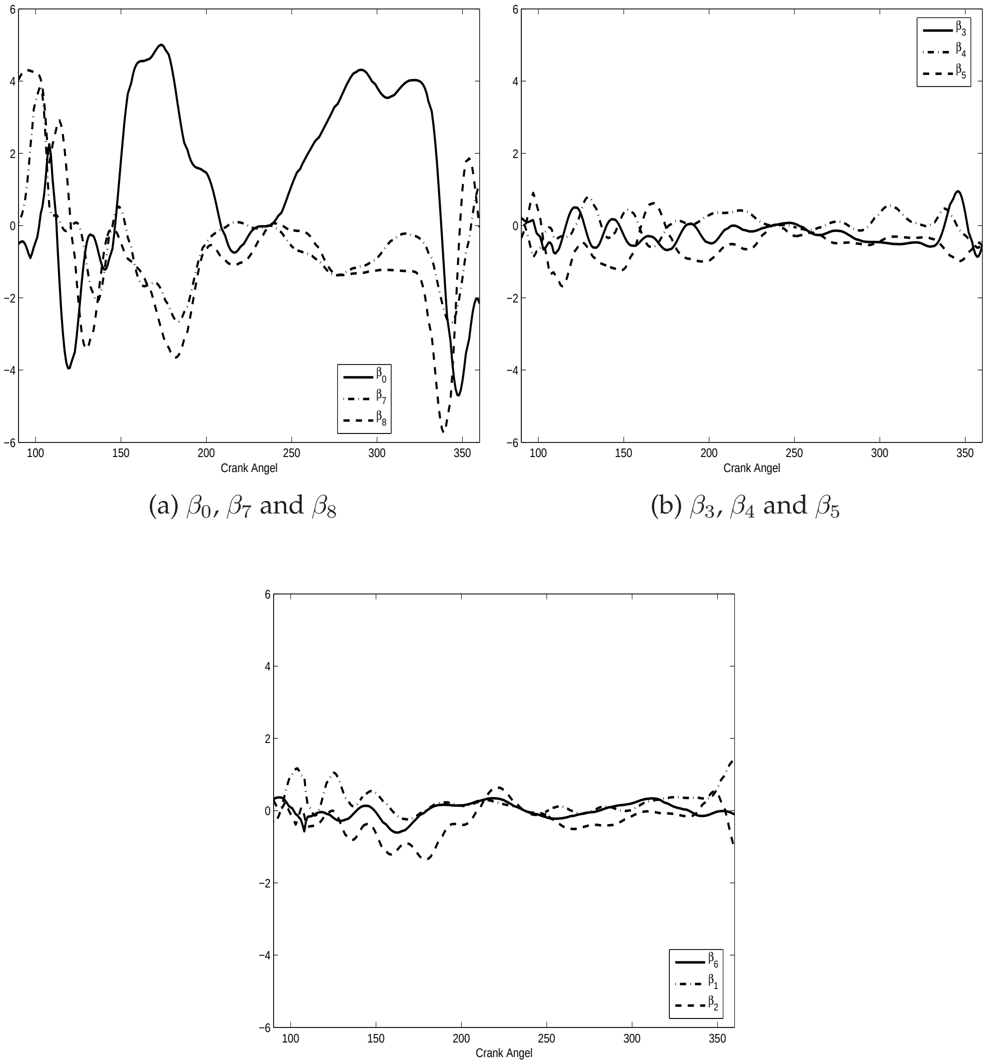
In this section, the valvetrain data of the crank angle within (90,360) is analyzed using semi-varying coefficient model.

Figure 5.9 give the OLS estimates of the functional coefficients for the valvetrain data within the crank angle (90,360). From these plots, we can see that only β_0, β_7 and β_8 are strongly depend on the crank angle. Hence we put these three coefficients into one group $\boldsymbol{\alpha}(u) = (\beta_0(u), \beta_7(u), \beta_8(u))$, and the rest of the coefficients into another group $\boldsymbol{\gamma} = (\beta_1, \beta_2, \cdots, \beta_6)$, which is treated as a constant coefficient vector. We also split the predictor $\mathbf{x}_i$ into two parts $\mathbf{w}_i$ and $\mathbf{z}_i$ which will corresponding to the different groups of the coefficients. Then we use the estimation procedure which is introduced and validated in previous sections to get the parameter estimates for model 5.7.

As in FLM, we still use Gaussian correlation function with single parameter as the correlation structure. θ is estimated by maximizing the penalized likelihood function together with the estimation for σ^2 and all the functional/constant coefficients. The detailed estimation procedure has been discussed in detail in previous sections. Figure 5.10 shows the MLE for θ and σ^2 along crank angle. The plot shows that parameter θ in correlation function varies between 1.4 and 2 along crank angle. Note that the correlation will become stronger if this parameter is small. This result says the correlation between the design cases are changing along crank angle. This figure also shows the MLE for σ^2 versus crank angle. The result shows that the MSE varies a lot along crank angle. We can also observe that when σ^2 achieves its peak, the estimated θ goes to the bottom.

The back-fitting result gives:

$$\hat{\gamma} = (\hat{\beta}_1, \cdots, \hat{\beta}_6) = (-0.2184, 0.0392, -0.5147, -0.0006, 0.2409, -0.2857).$$



(a) β_0, β_7 and β_8

(b) β_3, β_4 and β_5

(c) β_6, β_1 and β_2

Figure 5.9. *OLS estimates of all the functional coefficients within crank angel region (90,360)*

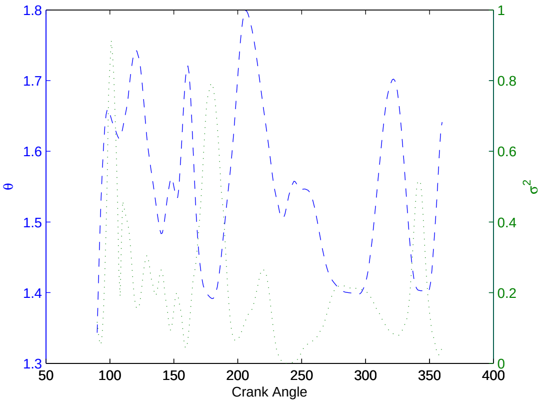
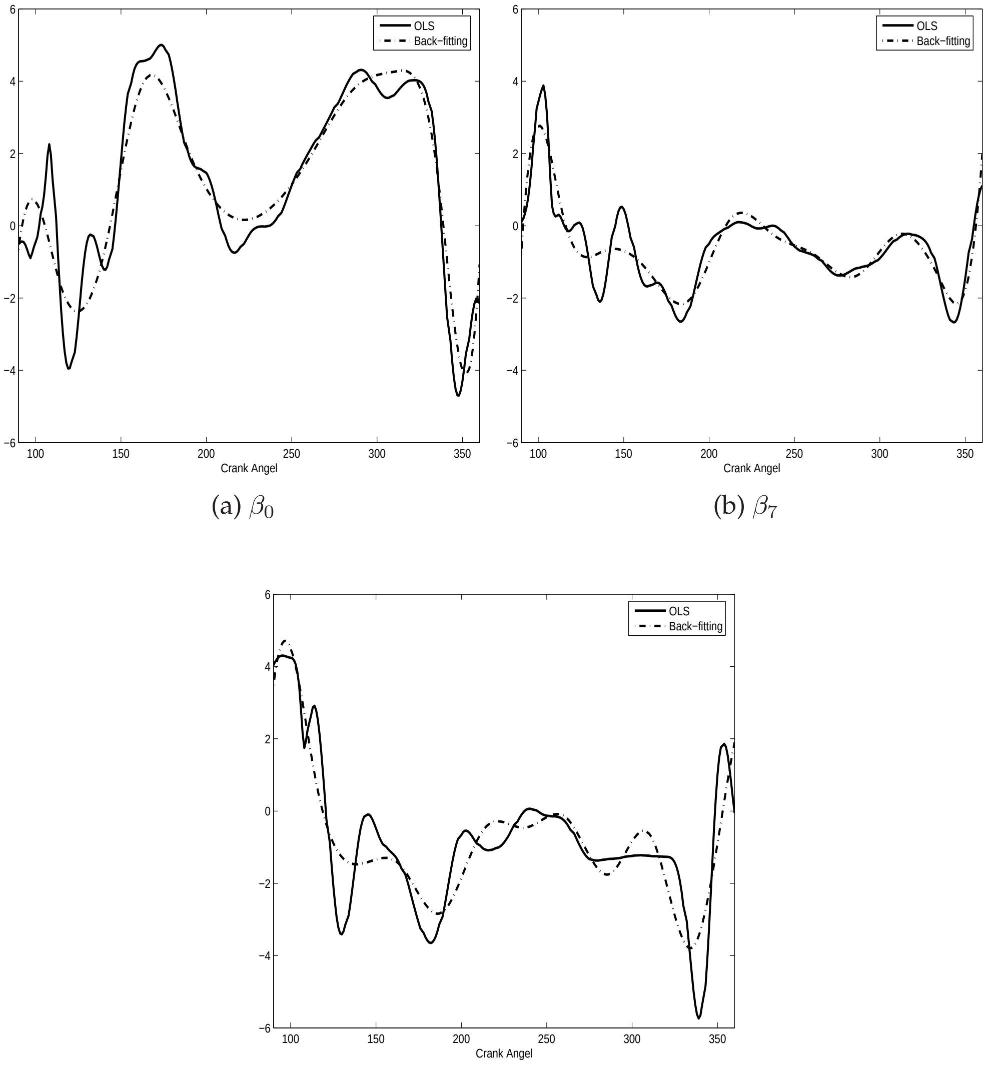


Figure 5.10. *MLE for parameter θ and σ^2 in model 5.7*

The result tells us that overall, x_1, x_3, x_4, and x_6 will try to decrease the valve motion error, while x_2 and x_5 will try to increase the motion error. The estimated varying coefficients $\beta_0(u)$, $\beta_7(u)$ and $\beta_8(u)$ from back-fitting algorithm and OLS are given in figure 5.11. Figures 5.12 to 5.13 give the SAE simulated motion error and the OLS and back-fitting algorithm for semi-varying coefficient model result for all the 16 design cases. From these plots, we can see the restored information from back-fitting algorithm for the semi-varying coefficient is smoother than the OLS estimates. But also there are a certain degree of information distortion for this estimating procedure as seen from these plots.

Figure 5.14 shows the Sobol' indices for the linear effect in model 5.7. Note that here, w_1 and w_2 corresponding to x_7 and x_8, z_j corresponding to x_{j+2} for $j = 1, \cdots, 6$. From this plot, we can see that the linear effect of variable x_7 and x_8 (w_1 and w_2) explains most of the total variance except at some crank angle region. Around crank angle 150, the linear effects of x_5 (z_5) explain almost 50% of the total variance, while the linear effect of x_7 and x_8 almost explain none of the total variance. Around crank angle 120, the linear effect of x_7 drops to almost zero, and x_8 drops to about

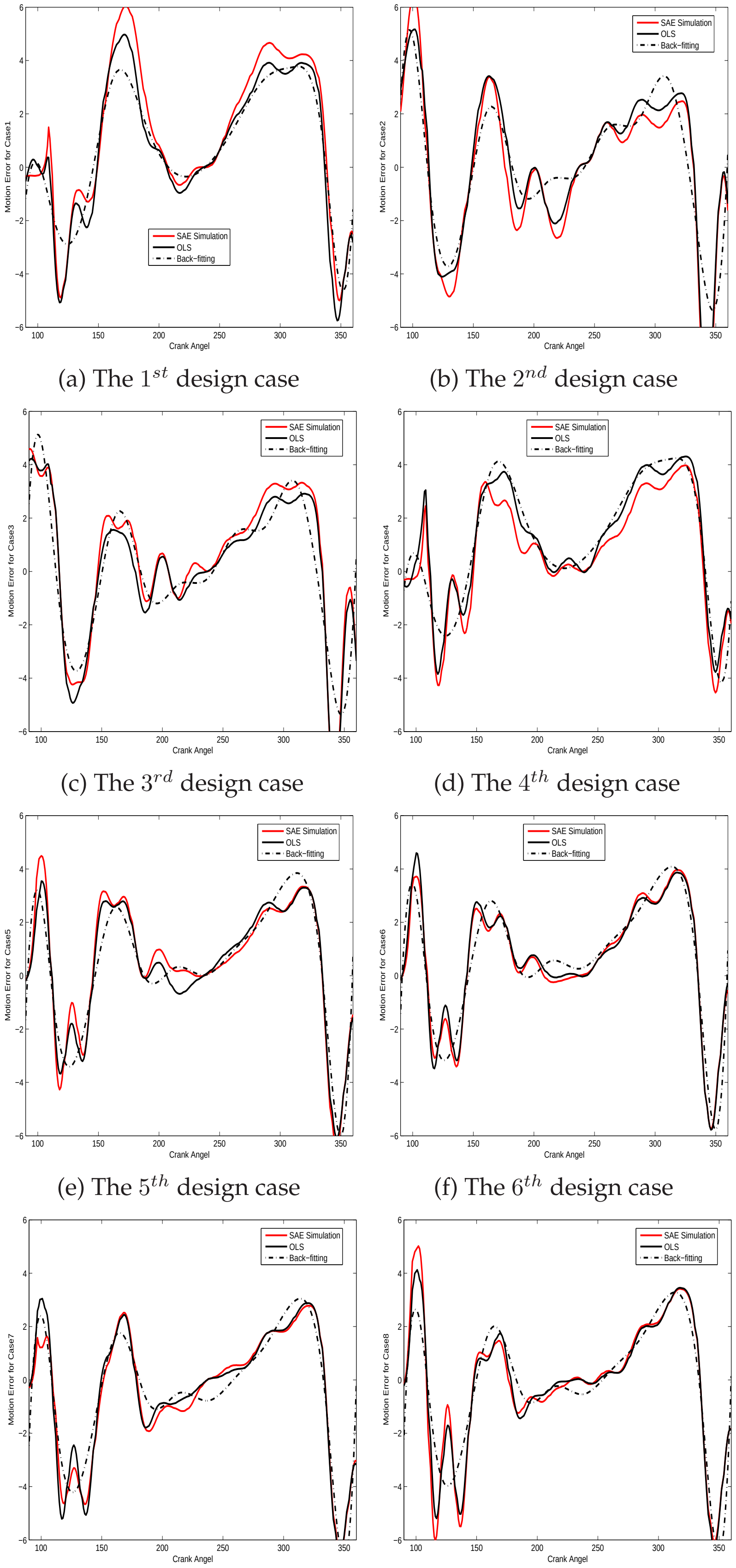


(a) β_0

(b) β_7

(c) β_8

Figure 5.11. *Back-fitting and OLS estimates of the functional coefficients β_0, β_7 and β_8 within crank angel region (90,360)*



(a) The 1^{st} design case

(b) The 2^{nd} design case

(c) The 3^{rd} design case

(d) The 4^{th} design case

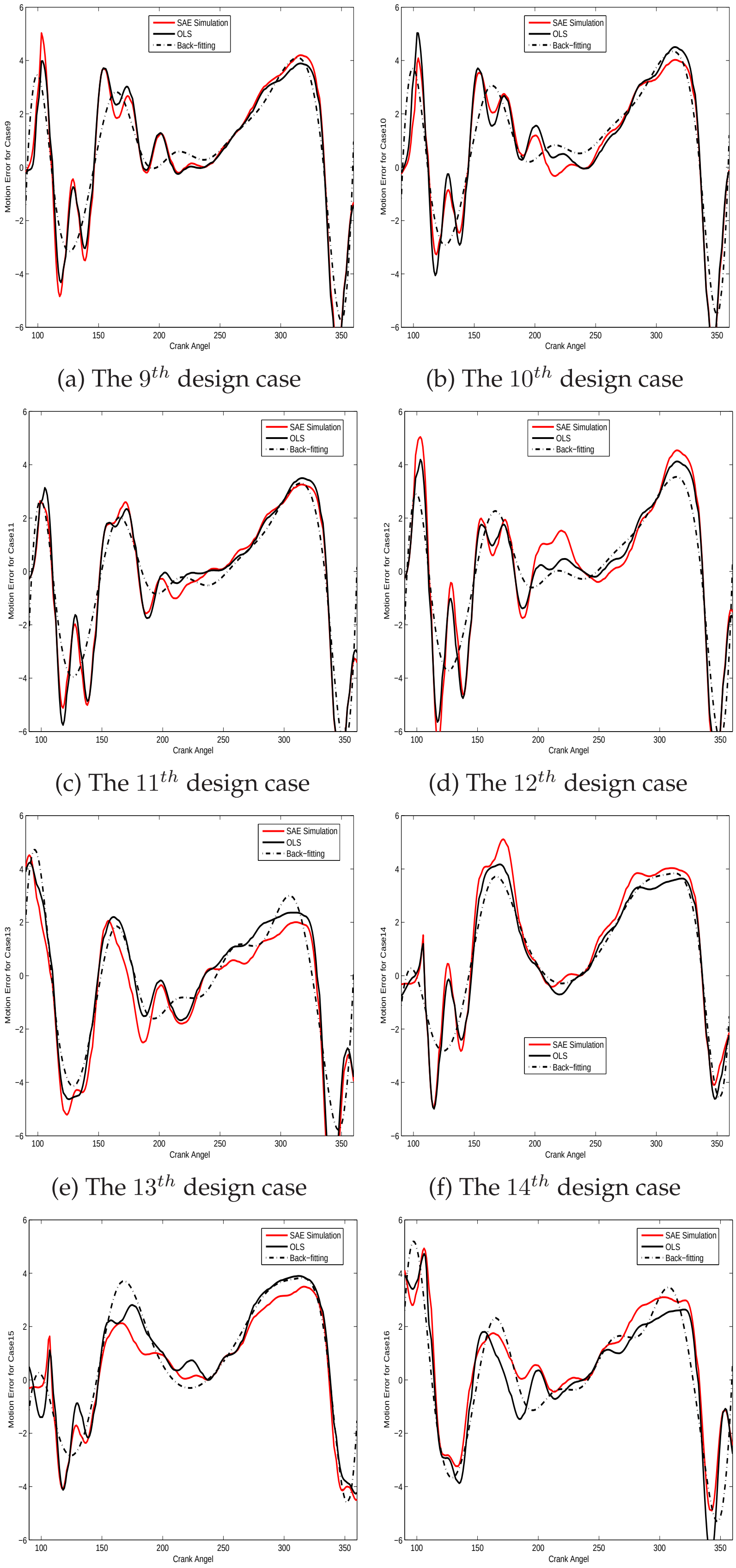
(e) The 5^{th} design case

(f) The 6^{th} design case

(g) The 7^{th} design case

(h) The 8^{th} design case

Figure 5.12. *Estimated valvetrain motion errors with OLS and Back-fitting algorithm for the first 8 design cases in Table 4.1 within crank angel region (90,360)*



(a) The 9^{th} design case

(b) The 10^{th} design case

(c) The 11^{th} design case

(d) The 12^{th} design case

(e) The 13^{th} design case

(f) The 14^{th} design case

(g) The 15^{th} design case

(h) The 16^{th} design case

Figure 5.13. *Estimated valvetrain motion errors with OLS and Back-fitting algorithm for the last 8 design cases in Table 4.1 within crank angel region (90,360)*

20% of the total variance. Instead, the linear effect of x_1 and x_5 explains about 20% of the total variance each.

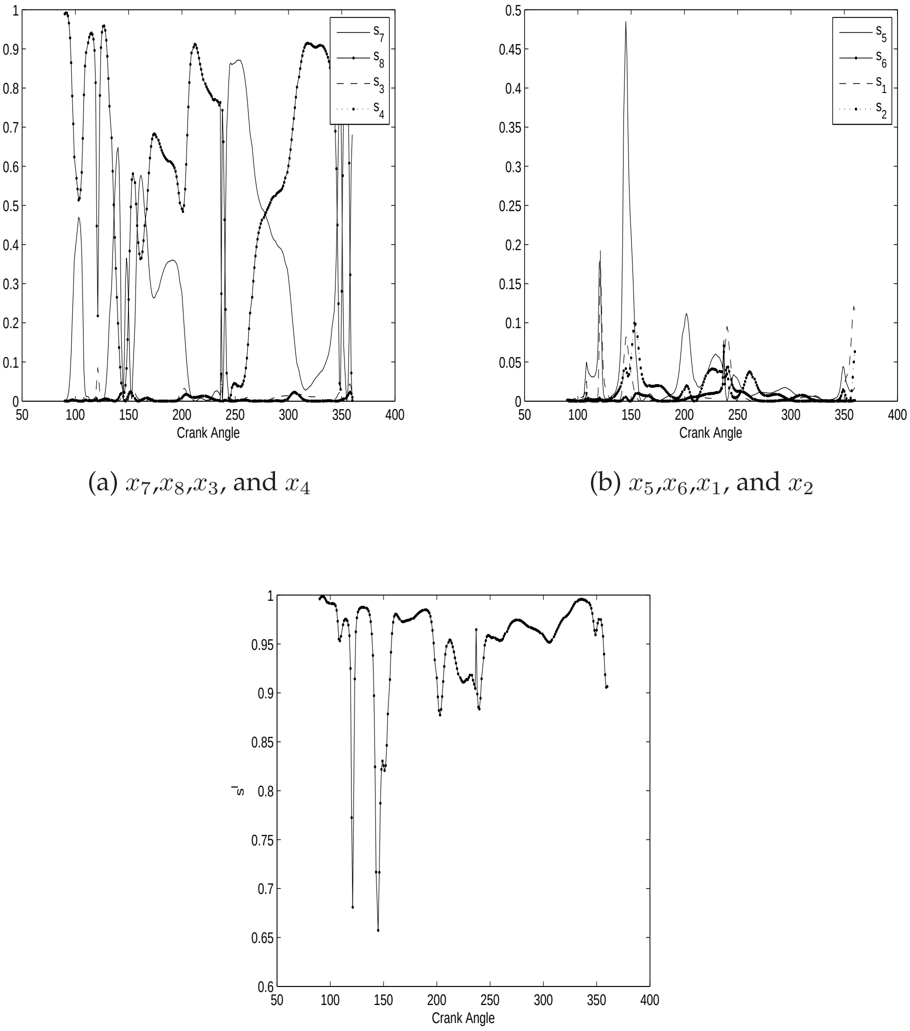
The third plot of figure 5.14 shows the sum of linear effect for all the variables. We can see that except for crank angle around 120 and 150, the linear effect explains over 90% of the total variance. The dramatic drop around crank angle 120 and 150 indicates that the effect of interaction between those variables explains more around those regions.

The above figures only give us information on how much each/total linear effect explain the total variance. To better understand how each variable affects the response, we need to get the estimation for each component function. After the functional ANOVA decomposition, each component g_i is a function of crank angle and the corresponding variable x_i which has been normalized and the range is in $(0, 1)$. We can calculate g_i at any given grid set on $(90, 360)$ by $(0, 1)$ to get an estimate of the component surface. Figures (5.15) and (5.16) show the contour plots for g_1 to g_8. We can see clearly how does each component function vary on the predictor domain.

5.8 Conclusion

In this chapter, we consider a spatial-temporal model with the overall trend by a partially functional linear model. In practice, some functional coefficients in functional linear model may be modeled as a constant. In the consideration of prediction accuracy, it is desirable to obtain a parsimonious model. This is the motivation we consider partial functional linear model to represent the overall trend. The backfitting algorithm is proposed to estimate both constant and functional coefficients in PFLM.

We proposed a new estimation procedure for the spatial-temporal model with partially functional linear model trend. First, estimate the parameters in Gaussian

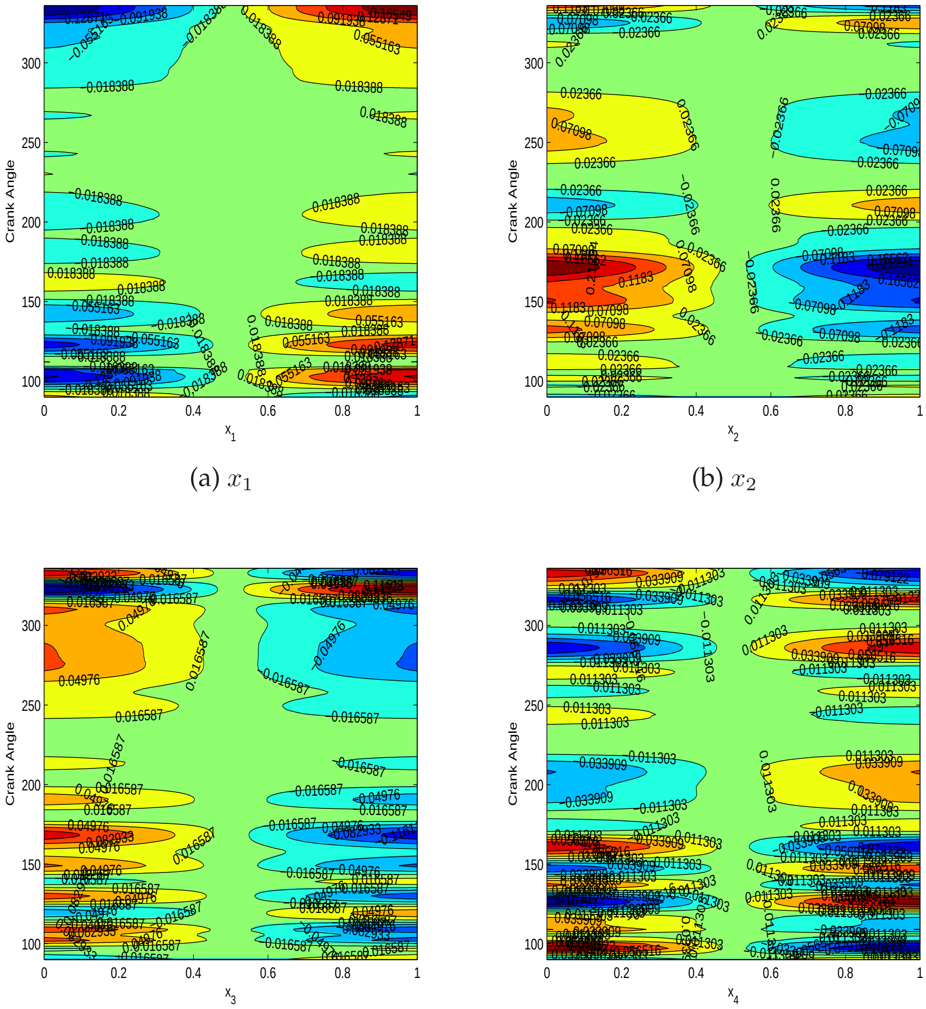


(a) x_7,x_8,x_3, and x_4

(b) x_5,x_6,x_1, and x_2

(c) Sum of Sobol′ indices for all linear effects

Figure 5.14. *Sobol′ indices for linear effects for crank angel within (90,360) for model 5.7*



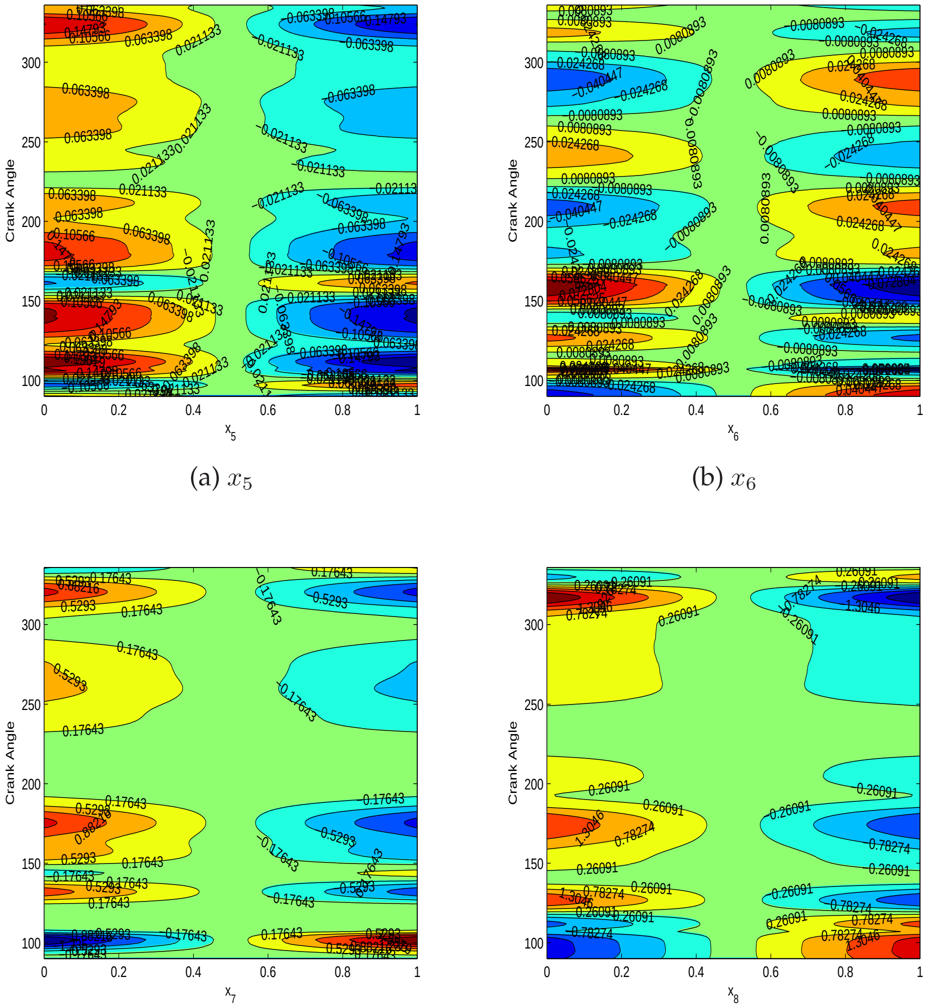
(a) x_1

(b) x_2

(c) x_3

(d) x_4

Figure 5.15. *Contour plots for x_1 to x_4*



(a) x_5

(b) x_6

(c) x_7

(d) x_8

Figure 5.16. *Contour plots for x_5 to x_8*

correlation function by treating the model as a general FLM using the estimation procedure proposed in Chapter 4. Then we use backfitting algorithm to estimate the constant and functional coefficients by treating the variance and variance-covariance matrix as the estimated ones from the FLM. The functional ANOVA decomposition is also a natural extension of the ones for FLM.

We conduct simulation studies to assess the proposed methodology. From our simulation, the proposed estimation procedures work well. Furthermore, we conduct a simulation to compare the prediction accuracy for multivariate kriging with the one used in Fang, Li and Sudjianto (2005). The result clearly shows the advantage of our proposed procedure over Fang, Li and Sudjianto (2005).

We then applied the proposed model and estimation procedure to valvetrain example for crank angle in $(90, 360)$. From the estimated functional coefficients in FLM, we can see clearly that the functional intercept and functional coefficients for design variable x_7 and x_8 have significant larger variation range than all other functional coefficients. Hence we use PFLM in this situation. The result shows that both the variance and the correlation between design cases are changing along crank angle direction. We also observed that when variance goes to the peak, the correlation will become stronger. The estimated constant coefficients tells us that overall, x_1, x_3, x_4, and x_6 will try to decrease the valve motion error, while x_2 and x_5 will try to increase the motion error.

From the result of functional ANOVA decomposition, we can see that the linear effect of variable x_7 and x_8 explains most of the total variance except at some crank angle region. Around crank angle 150, the linear effects of x_5 (z_5) explain almost 50% of the total variance, while the linear effect of x_7 and x_8 almost explain none of the total variance. Around crank angle 120, the linear effect of x_7 drops to almost zero, and x_8 drops to about 20% of the total variance. Instead, the linear effect of x_1

and x_5 explains about 20% of the total variance each. Except for crank angle around 120 and 150, the linear effect explains over 90% of the total variance. The dramatic drop around crank angle 120 and 150 indicates that the effect of interaction between those variables explains more around those regions.

Again, the methodology in this chapter can be applied directly to any computer experiments with functional coefficients in FLM have different degree of variation. The valvetrain example is just one of the real cases for such computer experiments.

Chapter 6

Final Remarks

6.1 Summary

Most existing modeling procedures for computer experiments in the literature can only deal with single response computer experiments. Motivated by two examples shown in Chapter 1 and analyzed in Chapters 3, 4 and 5, we systematically studied how to model functional response computer experiments.

We first studied how to model computer experiments with multiple responses since this is closely related to model sparse functional response computer experiments. We proposed a multivariate kriging model for multiple response computer experiments. The resulting predictor of the multivaraite kriging model is the best linear unbiased predictor and automatically interpolates the observed data. Since the multivariate kriging model takes into account the correlation between multiple responses, it provides us more accurate prediction than modeling the multiple response one by one using univariate kriging model. The accuracy gain may be up to 30% for high noise level. Using theory of matrix normal distribution, we proposed an estimation procedure for the multivariat kriging model and developed an algorithm to obtain the estimate of model parameters. To investigate the impact of each variable, we developed an multivariate version of functional ANOVA decomposition for the multivariate kriging model. We further derived the closed forms of each function for main effect. Closed forms for total variance and variance for each main effect are also derived. By doing functional ANOVA decomposition on the multivariate kriging model, we can get the clear picture of the shape of each component function over the functional and design domain, after removing the effect of all other variables. This decomposition is highly useful to answer the questions such as how a particular variable affects the multiple responses, how important a specific variable is to the multiple responses, etc.

In Chapters 4 and 5, we developed spatial-temporal models for functional response computer experiments, and demonstrated how to apply them to model computer experiments. We consider both functional linear models and partially functional linear models to represent thee overall trend in the proposed spatial-temporal models. Based on the data structure of functional response computer experiments, we proposed estimation procedures for the functional linear models and partially functional linear models. We further developed several method for prediction of functional response computer experiments. Compared with the one proposed in Fang, Li and Sudjianto (2005), our proposed procedure yields more accurate prediction.

We also constructed the functional ANOVA decomposition for spatial-temporal model with the overall trend being either partially functional linear model or functional linear model. The closed forms for each linear individual component are obtained. The decomposition is very useful to examine the effects of variables of interest on the functional response.

6.2 Future Works

In this dissertation, we focus on functional response with 1-dimensional index. It is certain of interest to study functional response with multi-dimensional index. Image data can be viewed as functional response with 2-dimensional index.

Object data or tree-structure data may be regarded as functional response with 3-dimensional index. Future works are need to conduct research on how to analyze image functional data and object functional data.

In this dissertation, we consider only a simple situation of functional response computer experiment. In practice, a simulation systems may be very complicated and consist of multilevel of sub-computer experiments. It is of interest to extend this dissertation work to a complex, multilevel and functional response computer experiments.

Bibliography

An, J. and Owen, A. B. (2001). Quasi-regression. *Journal of Complexity*, 17:588–607.

Antoniadis, A. and Fan, J. (2001). Regularization of wavelets approximations (with discussions). *Journal of American Statistics Associate*, 96:939–967.

Arnold, S. F. (1981). *The Theory of Linear Model and Multivariate Analysis*. Wiley, Newyork.

Association, A. P. (1980). *Diagnostic and Statistical Manual of Mental Disorders*. American Psychiatric Association, Washington, DC.

Bates, R., Giglio, B., and Wynn, H. (2003). A global selection procedure for polynomial interpolators. *Technometrics*, 45:246–255.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford.

Breitung, K. (1984). Asymptotic approximationsfor multinomial integrals. *Journal of the Engineering Mechanics Division*, 110:357–367.

Cardot, H., Ferraty, F., and Sarda, P. (1999). Spline estimators for the functional linear model: consistency, application and splus implementation. Technical Report 1999/1, Rapports Techniques Toulouse.

Cressie, N. (1986). Kridging non-stationary data. *Journal of American Statistics Assoc.*, 81:625–634.

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

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

Cutler, A. and Cordero-Brana, O. I. (1996). Minimum hellinger distance estimation for finite mixture models. *Journal of the American Statistical association*, 91:1716–1743.

Cybenko, G. (1989). Approximation by superposition of a sigmoidal function. *Math. Contr. Signal Systems*, 2:303–314.

De Door, C. (1978). *A practical Guide to Splines*. Springer, Berlin.

Du, X. and Sudjianto, A. (2004). The first order saddlepoint approximation for reliability analysis. *AIAA*, 42(6):1199–1207.

Eilers, P. H. and Marx, B. D. (1996). Flexible smoothing with b-splines and penalties. *Statistical Science*, 11(2):89–102.

Fan, J. (1993). Local linear regression smoothers and their minimax efficiency. *Annals of Statistics*, 21:196–216.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modeling and Its Applications*. Chapman and Hall, London.

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

Fan, J. and Li, R. (2004). New estimation and model selection procedures for semiparametric modeling in longitudinal data analysis. *Journal of American Statistical Association*, 99:710–723.

Fan, J. and Zhang, J.-T. (2000). Two-step estimation of functional linear models with applications to longitudinal data. *Journal of the royal statistical society, series B(Statistical methodology*, 62(2):303–322.

Fan, J. and Zhang, W. (1999). Statistical estimation in varying coefficient models. *The Annals of Statistics*, 27(5):1491–1518.

Fang, K., Li, R., and Sudjianto, A. (2005). *Design and Modeling For Computer Experiments*. Chapman and Hall, New York.

Fang, K., Lin, D., Winker, P., and Zhang, Y. (2000). Uniform design: Theory and applications. *Technometrics*, 42:237–248.

Faraway, J. J. (1997). Regression analysis for a functional response. *Technometrics*, 39(3):254–262.

Ferraty, F. and Vieu, P. (2002). The functional nonparametric model and application to spectrometric data. *Computational Statistics*, 17(4):545–564.

Frank, I. and Friedman, J. (1993). A statistical view of some chemometrics regression tools. *Techometrics*, 35:109–148.

Friedman, J. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, 19:1–141.

Gérard, F., Tournour, M., El Masri, N., Cremers, L., Felice, M., and Selmane, A. (2001). Numerical modeling of engine noise radiation through the use of acoustic transfer vectors - a case study.

Giglio, B., Riccomagno, W., and Wynn, H. (2000). Gróbner bases strategies in regression. *Journal of Applied Statistics*, 27:923–938.

Hasofer, A. and Lind, N. (1974). Exact and invariant second-moment code format. *Journal of the Engineering Mechanics*, 100:111–121.

Hassoun, M. (1995). *Fundamentals of Artificial Neural Networks*. The MIT Press, Cambridge, Massachusetts.

Hastie, T. and Mallows, C. (1993). A discussion of A statistical view of some chemometrics regression tools by I.E. Frank. and J.H. Friedman. *Technometrics*, 53:140–143.

Hastie, T. and Tibshrani, R. (1993). Varying-coefficient models. *Journal of the Royal Statistical Society, Series B(Methodological)*, 55(4):757–796.

Haykin, S. (1998). *Neural Networks: A comprehensive Foundation*. Prentice Hall, Upper Saddle River, NJ.

Hoover, D. R., Rice, J. A., Wu, C. O., and Yang, L.-P. (1998). Non-parametric smoothing estimates of time-varying coefficient models with longitudinal data. *Biometrica*, 84(4):809–822.

Hϕst, G. (1999). Kridging by local polynomials. *Computational Statistics and Data Analysis*, 29:295–312.

Jones, M., Marron, J., and Sheater, S. (1996a). A brief survey of bandwidth selection for density estimation. *Journal of American Statistics Associate*, 91:401–407.

Jones, M., Marron, J., and Sheater, S. (1996b). Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, 11:337–381.

Journel, A. and Huijbregts, C. (1978). *Mining Geostatistics*. Academic Press, London.

Kömer, T. (1988). *Fourier Analysis*. Cambridge University Press.

Kim, B. (1984). *Studies of multinomial mixture models*. PhD thesis, University of North Carolina - Chapel Hill.

Kleijnen, J. (1987). *Statistical Tools for Simulation Practitioners*. Marcel Decker, New York.

Koehler, J. and Owen, A. (1996). *Handbook of Statistics*. Elsevier Science B.V.

Laslett, G. M. (1994). Kriging and splines: An empirical comparison of their predictive performance in some applications. *Journal of the American Statistical Association*, 89:391–400.

Lehman, J. (2002). *Sequential Design of Computer Experiments for Robust Parametric Design*. PhD thesis, Department of Statistics, Ohio State University, Columbus, OH.

Li, R. and Sudjianto, A. (2005). Analysis of computer experiments using penalized likelihood in Gaussian Kriging models. *Technometrics*, 47:111–120.

Li, R., Sudjianto, A., and Zhang, Z. (2005). Modeling computer experiments with functional response. In *SAE Paper, No. 2005-01-1397*, Detroit, Michigan, USA.

Lin, D. and Ying, Z. (2001). Semiparametric and nonparametric regression analysis of longitudinal data(with discussion). *Journal of American Statistical Association*, (96):103–126.

Loan, V. and F., C. (1997). *Introduction to Scientific Computing*. New Jersey: Prentice Hall.

MacKay, D. (1998). Introduction to gaussian processes, in c.m. bishop, ed. *'Neural Networks and Machine Learning', NATO ASI Series, Kluwer*, pages 133–166.

MacKay, M. (1995). Evaluating prediction uncertainty. NU REG/CR-6311, Nuclear Regulatory Commission and Los Alamos National Laboratory.

Mardia, K. and Marshall, R. (1984). Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135–146.

Marron, J. and Nolan, D. (1988). Canonical kernels for density estimation. *Statisti. Prob. Lett.*, 7:195–199.

Marx, B. and Eilers, P. (1996). Generalized linear regression on sampled signals with penalized lekelihood. In Forcina, A., Marchetti, G. M., Hatzinger, R., and Galmacci, G., editors, *Proceeding of the 11th International workshop on statistical modelling*, Orvieto. Statistical Modelling.

Matern, B. (1947). Method of estimating the accuracy of line and sample plot survey. *Medd. Skogsforskn Inst.*, 36(1).

Matheron, G. (1963). Principles of geostatistics. *Econom. Geol.*, 58:1246–1266.

Melchhers, R. (1999). *Structural Reliability Analysis and Prediction.* John Wiley and Sons, Chichester, England.

Moyeed, R. and Diggle, P. (1994). Rates of convergence in semiparametric modeling of longitudinal data. *Austr. Jour. Statist.*

Nadaraya, E. (1964). On estimating regression. *Theory Prob. Appl.*, 9:141–142.

O'Hagan, A. (1989). Comment: Design and analysis of computer experiment. *Statist. Sci.*, 4(4):430–432.

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems(with discussion). *Statistical Science*, 1:505–527.

O'Sullivan, F. (1988). Fast computation of fully automated log-density and log-hazard estimators. *SIAM Journal of Sci. Statist. Comput.*, 9:363–379.

Owen, A. R. (1988). Computing empirical likelihoods. *Computing Science and Statistics, Procedings of the 20th Symposium on the Interface*, pages 442–447.

Patterson, D. (1996). *Artificial Neural Networks.* Singapore: Prentice Hall.

Philips, P. J., Schamel, A. R., and Meyer, J. (1989). An efficient model for valvetrain and spring dynamics.

Ramirez, R. W. (1985). *The FFT: Fundamentals and Concepts*. Englewood Cliffs, NJ: Prentice-Hall.

Ramsay, J. and Silverman, B. (1997). *Functional Data Analysis*. Springer-Verlag.

Rao, C. (1973). *Linear Statistical Inference and its Applications*. Wiley, New York.

Ripley, B. (1981). *Spatial Statistics*. Wiley, New York.

Rumelhart, D., Hinton, G., and Williams, R. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA.

Ruppert, D. (2002). Selecting the number of knots for penalized splines. *Journal of Computational&Graphical Statistics*, 11:735–757.

Ruppert, D. and Carroll, R. J. (1999). A simple roughness penalty approach to regression spline estimation. Technical Report 1249, School of OR&IE, Cornell University.

Ruppert, D., Sheather, S., and Wand, M. (1995). An effective bandwidth selector for local least squares regression. *Journal of American Statistical Association*, 90:1257–1270.

Sacks, J., Schiller, S., and Welch, W. (1989a). Design for computer experiments. *Technometrics*, 31(1):41–47.

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

Santner, T., Williams, B., and Notz, W. (2004). *The Design and Analysis of Computer Experiments*. Springer.

Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K. (2001). Metamodels for computer-based engineering design: survey and recommendations. *Engineering with Computers*, 17:129–150.

Sobol', I. (1993). Sensitivity analysis for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment*, 1:407–414.

Sobol', I. (2001). Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Math. and Comp. in Simulation*, 55:271–280.

Sobol', I. (2003). Theorems and examples on high dimensional model representation. *Reliability Engineering & System Safety*, 79:187–193.

Stein, M. (1989). Comment: Design and analysis of computer experiments. *Statist. Sci*, 4(4):432–433.

Stone, C., Hansen, M., Kooperberg, C., and Truong, Y. (1997). Polynomial splines and their tensor products in extended linear models. *Annals of Statistics*, 25:1371–1470.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.*, B.58:267–288.

Tu, J. and Jones, D. (2003). Variable screening in metamodel design by crossvalidated moving least square method. *44th AIAA Structure Dynamics and Material Conference*, pages 1669–1673.

Wahba, G. (1990). *Spline models for observational data*, volume 59. CBMS-NSF Regional Conference series in applied mathematics, Philadelphia, Pennsylvania, 16th edition.

Watson, G. (1963). Smooth regression analysis. *Sankhya Ser.*, A26:359–372.

Wu, C. O., Chiang, C.-T., and Hoover, D. R. (1998). Asymptotic confidence regions for kernel smoothing of a varying-coefficient model with longitudinal data. *Journal of the American Statistical Association*, 93(444):1388–1402.

Yaglom, A. (1987). *Correlation Theory of Stationary and Related Random Functions*, volume 1. Springer, New York.

Zeger, S. and Diggle, P. (1994). Semiparametric models for longitudinal data with application to cd4 cell numbers in hiv seroconverters. *Biometrics*, (50):689–699.

Zhang, Z. (2005). Modeling computer experiments with functional response using non-parametric and semi-parametric models. Master's thesis, Department of Statistics, Pennsylvania State University, University Park, PA.

Vita

Zhe Zhang

Education

The Pennsylvania State University State College, Pennsylvania June, 2005–May, 2007

Ph.D. in Statistics, expected in May, 2007
Area of Specialization: Computer experiments, kriging model, spatial-temporal model

The Pennsylvania State University State College, Pennsylvania August, 2003–May, 2005

Master in Statistics, expected in May, 2005
Area of Specialization: Functional Linear Model

Tianjin University Tianjin, P.R. China 1997–1999

Ph.D. in Engineering Mechanics,
Area of Specialization: Fluid Dynamics and Computational Fluid Dynamics

Tianjin University Tianjin, P.R. China 1994–1997

M.S. in Engineering Mechanics,
Area of Specialization: Fluid Dynamics and Computational Fluid Dynamics

Tianjin University Tianjin, P.R. China 1990–1994

B.S. in Engineering Mechanics

Working Experience

Quantitative Finance Analyst Vice President, Global Risk, Bank of America August, 2006–Present

Risk Analyst III Vice President, Global Anti-Money Laundering Operation, Bank of America March, 2006–August, 2006

Summer Intern Global Quality and Productivity, Bank of America April, 2005–September, 2005