HETEROSCEDASTIC GAUSSIAN PROCESS REGRESSION USING NEAREST NEIGHBOR POINT ESTIMATES

THESIS
This thesis is submitted
as partial fulfillment for Master degree
from Institut Teknologi Bandung

by
Muhammad Daffa Robani
23619020
(Aerospace Engineering Master Program)

Institut Teknologi Bandung
2021

Legalisation Page

HETEROSCEDASTIC GAUSSIAN PROCESS REGRESSION USING NEAREST NEIGHBOR POINT ESTIMATES

by

Muhammad Daffa Robani

23619020

Aerospace Engineering Master Program

Institut Teknologi Bandung

Bandung, 18 January 2021

Approved by:

Advisor:

Lavi R, Zuhal

Dr. Lavi Rizki Zuhal

19730116 200604 1 001

Co-Advisor:

Dr. Pramudita Satria Palar

19880622 201212 1 001

Abstract

HETEROSCEDASTIC GAUSSIAN PROCESS REGRESSION USING NEAREST NEIGHBOR POINT ESTIMATES

by

Muhammad Daffa Robani

23619020

(Aerospace Engineering Master Program)

In exploring engineering design based on physical experiments/stochastic simulators, noise might have significant effects on the observed sample that the use of deterministic surrogate model is not inadequate since its inability to distinguish the true function and the noise. Gaussian process (GP) is one type of surrogate models that have been extensively used in practice to handle noisy problems. In its standard procedure, the standard homoscedastic GP learn the noisy samples by assuming the noise level is uniform across the input spaces. This variant of GP however is not appropriate when the variance of the noise is varying in the input spaces, i.e., heteroscedastic. Various heteroscedastic GP (HGP) models have been developed to extend the GP model in learning heteroscedastic problems. This study focuses on this HGP model that does not require any additional replication, which has advantages in terms of experimental cost. One of the latest approaches to this variant of HGP model is called Improved Most Likely Heteroscedastic GP (IMLHGP) that have already showed good performances in modeling heteroscedastic problems. IMLHGP, as the state-of-the-art, also has a very attractive computational efficiency in training the model. However, the IMLHGP model is found to require high number of training samples as the model is observed to perform poorly when the number of training samples is rather low. This requirement is definitely unfavorable when the experimental cost is already expensive, such as in aerospace problems. In reducing the training sample size requirement, this thesis proposes a modified HGP model based on the IMLHGP by augmenting an extra distance-based nonparametric regression to tackle the wrongly interpolating noise level predictions that frequently appears in IMLHGP when the number of training samples is small. This proposed model called Nearest Neighbor Point Estimates HGP (NNPEHGP) is tested on several heteroscedastic problems, consisting of two mathematical functions, two real-world experimental datasets, two real-world analytical problems, and a stochastic simulator. The results show the superiority of NNPEHGP to the IMLHGP in terms of predictive accuracy and robustness, especially when the number of training samples is low. It is also found that NNPEHGP is much more stable in learning high dimensional heteroscedastic problems.

Keywords: Gaussian processes surrogate model, heteroscedastic

Abstrak

MODEL REGRESI HETEROSKEDASTIK BERBASIS *GAUSSIAN PROCESS* DAN ESTIMASI DARI NILAI TETANGGA TERDEKAT

oleh
Muhammad Daffa Robani
23619020
(Program Studi Teknik Dirgantara)

Dalam mengeksplorasi desain menggunakan eksperimen laboratorium/simulator stokastik, penggunaan *surrogate model* deterministik seringkali tidak cukup karena ketidakmampuannya untuk mempertimbangkan pengaruh *noise* pada observasi yang didapatkan. *Gaussian process* adalah salah satu contoh *surrogate model* yang sudah sering digunakan dalam mempelajari masalah-masalah yang terdapat *noise* di dalamnya. Dalam prosedur standarnya, *standard homoscedastic* GP bekerja dengan mengasumsikan seragamnya kekuatan *noise* dalam ruang inputnya. Sehingga, varian GP ini tidak cukup untuk menangani kasus di mana kekuatan *noise*-nya bervariasi (heteroskedastik). Beberapa model *heteroscedastic* GP sudah dikembangkan untuk menangani masalah seperti ini. Studi pada tesis ini berfokusi pada model HGP yang tidak membutuhkan replikasi tambahan pada sampel yang diamati karena kelebihannya dari perspektif kebutuhan biaya eksperimen. *Improved Most Likely Heteroscedastic* HGP (IMLHGP), sebagai salah satu pendekatan terkini untuk model HGP, telah menunjukkan performanya yang baik dalam memodelkan masalah heteroskedastik. IMLHGP juga memiliki efisiensi komputasional yang sangat tinggi dalam proses pembuatan modelnya. Akan tetapi, IMLHGP ditemukan untuk membutuhkan jumlah sampel yang sangat tinggi dalam membuat modelnya, performanya akan sangat menurun ketika jumlah sampelnya sedikit. Kebutuhan akan jumlah sampel yang tinggi ini tidak menguntungkan saat biaya eksperimen untuk memperoleh sampel sudah terhitung mahal, seperti masalah-masalah dalam bidang kedirgantaraan. Dalam mengurangi kebutuhan jumlah sampel ini, tesis ini mengajukan model HGP baru berdasar dari model IMLHGP dengan menambahkan model regresi nonparametrik di dalam prosedurnya untuk menghindari masalah interpolasi yang keliru dalam memprediksi kekuatan *noise* – yang sering dialami IMLHGP saat jumlah sampelnya sedikit. Model baru yang dinamakan *Nearest Neighbor Point Estimates* HGP (NNPEHGP) ini telah diuji dalam beberapa masalah heteroskedastik, yang terdiri dari dua fungsi matematik, dua data eksperimen, dua masalah analitik, dan simulator stokastik. Hasil-hasil yang diperoleh menunjukkan keunggulan dari NNPEHGP dalam akurasi dan kestabilan prediksinya, terlebih di jumlah sampel yang rendah. Hasil-hasil ini juga menunjukkan bahwa NNPEHGP jauh lebih stabil dari IMLHGP dalam menangani masalah heteroskedastik berdimensi tinggi.

Kata kunci: Gaussian process surrogate model,heteroskedastik

Thesis User Guide

The citation to this thesis in Indonesian can be written as follows:

Robani, M. D. (2021): *Model Regresi Heteroskedastik Berbasis Gaussian Process dan Estimasi dari Nilai Tetangga Terdekat*, Tesis Program Magister, Institut Teknologi Bandung.

and in English as follows:

Robani, M. D. (2021): *Heteroscedastic Gaussian Process Regression using Nearest Neighbor Point Estimates*, Master's Program Thesis, Institut Teknologi Bandung.

This thesis was written using LaTeX, a part of TeX project.

Acknowledgement

I might have not reached this point in my academic life without all these presences around me throughout these years.

1. All praises and deep gratitude to Allah SWT, with all His grace and guidance.
2. My mother and my father, for all the supports and love.
3. My thesis advisors, Dr. Lavi R. Zuhal and Dr. Pramudita S. Palar, for their constant help and trust throughout all the process of working this thesis.
4. My fellow laboratory mates for all the discussions we had.
5. My friends for all the moments we shared.

Bandung, 18 January 2021

Muhammad Daffa Robani

Contents

List of Figures

List of Tables

Chapter I

Introduction

I.1 Research Background

In the exploration of the impact of certain design features on an engineering design of any scale, simulations, be it experimental or numerical, often require a significant budget due to the large number of design variables to be considered, as is the case with aerospace engineering problems. For example, obtaining the numerical solution of a wings' aerodynamic efficiency using computational fluid dynamics (CFD) simulations may take several hours depending on the complexity and desired accuracy. For a large number of considered design variables to explore, more simulations would be required to accurately capture the behavior of each feature i.e., the underlying true function, thus leading to a large cost.

In order to effectively reduce the cost in capturing the behavior of various design features, a surrogate model is used in place of the underlying true function. Surrogate models are essentially approximation models that, based on a finite number of observed samples, will learn/approximate the underlying function that describes the data the best. In the context of engineering design exploration, variations of surrogate models have been studied, such as: linear regression (Zhang et al., 2018), support vector regression (Andrés et al., 2012), neural network (Rai and Madavan, 2000), radial basis function (RBF) (Wang et al., 2015), Kriging/Gaussian processes (GP) (Palar and Shimoyama, 2017), etc. By utilizing surrogate model(s), the required number of samples/experiments could be reduced while maintaining the representation of the physics based on the information of the design explorations.

Regarding surrogate models, recently, there has been an increasing trend in the utilization of the Gaussian processes (GP) surrogate model. Whereby, the latter would provide both an estimation of the true function and its statistics, such as the mean and variance (predictive interval). With this ability, the GP model has performed well in different areas of research, such as: design exploration (Clark Jr et al., 2016; Kanazaki et al., 2006), uncertainty quantification (Bhattrai, Baar, and Neely, 2018; Mukhopadhyay et al., 2017), and sequential optimization (Huang et al., 2006; Sun et al., 2014).

To differentiate between the two approaches in performing an experiment, namely, a physical real-time experiment and a numerical experiment/simulation, the data from the latter may be modelled without the considerations of present random variables or any randomness yielding it to be deterministic. Though numerical simulations may be stochastic i.e., contains numerical errors or the simulation itself is stochastic, this is contrast to the stochastic nature that comes from the noisy data of the physical experiments which may be significant in making complex, the process of learning the underlying true function. The GP surrogate model has been extensively used in modeling deterministic computer simulations in various engineering applications (Laurenceau and Sagaut, 2008; Mukhopadhyay et al., 2017; Weinmeister, Gao, and Roy, 2019). However, with the power of GP in making the statistical output available, this needs to be taken advantage from with the use of a noisy data set from a physical experiment as with Chaudhuri et al. (2015). The use of a conventional GP in physical experiments or the stochastic simulators may result in a surrogate model that interpolates the noise (see Fig. I.1), which is an undesirable situation. Therefore, efforts are needed to develop an appropriate GP model for physical experiments and stochastic simulators.

There have been several efforts to develop GP models that are proper for modeling noisy and random data. For example, the standard homoscedastic GP/regression Kriging (RK) is developed by assuming the noise level to be constant across the input space. This model treats noise in the samples by adding a constant noise parameter to the diagonal of the covariance matrix. The homoscedastic GP's prediction would unlikely interpolate the observed samples, which is similar to the regression analysis. The homoscedastic GP model has been used in efficient global optimization (Picheny, Wagner, and Ginsbourger, 2013), and yields accurate predictions on problems with uniform noise level i.e., homoscedastic noise (Forrester,

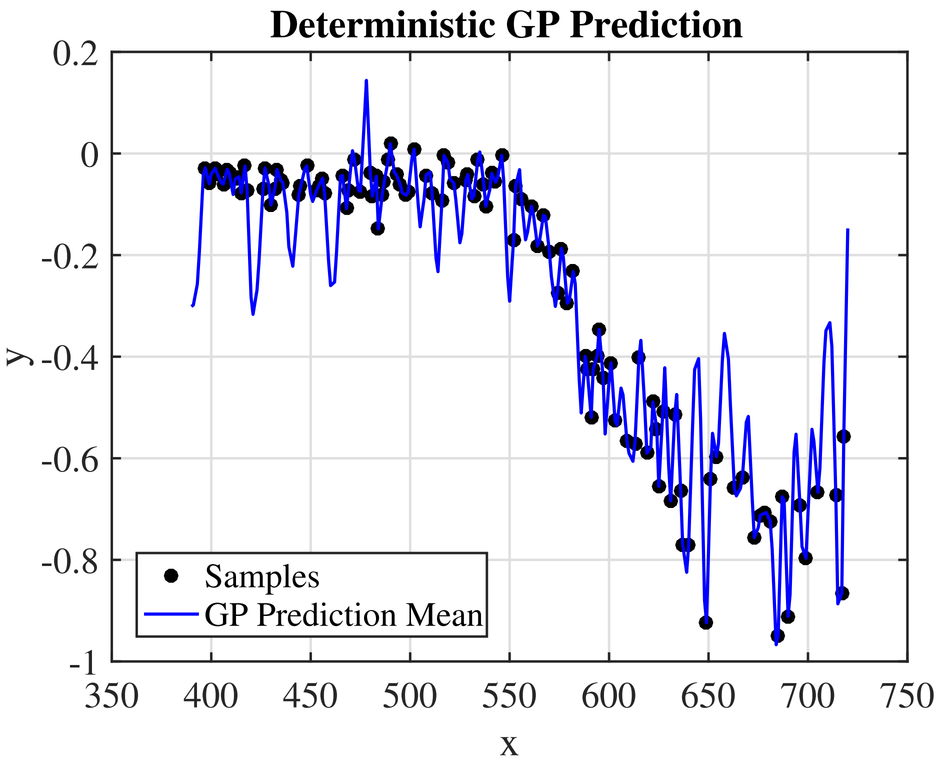


Figure I.1: Deterministic GP Prediction on Noisy Data.

Keane, and Bressloff, 2006). However, the noise level in some problems might vary across the input spaces (i.e., heteroscedastic noise). The GP's prediction distributions over the responses would most likely be incorrect if the standard homoscedastic GP model is employed in the heteroscedastic case. As visualized in Fig. I.2, the standard homoscedastic GP could not capture the smaller noise variances and higher noise variances in the leftward and the rightward regions of the input space, respectively. Even though the mean of the responses seems to be successfully captured by the standard homoscedastic GP, its varying noise level fails to be learned. For several engineering problems, the prediction on the noise level is equally as important since the designer may want to know which design has a small or high variance, such as in the context of uncertainty quantification and robust optimization. Therefore, the GP model is further developed by defining a more flexible assumption to its noise level for a more proper utilization in modelling heteroscedastic problems.

Various heteroscedastic GP (HGP) models have been developed for this purpose.

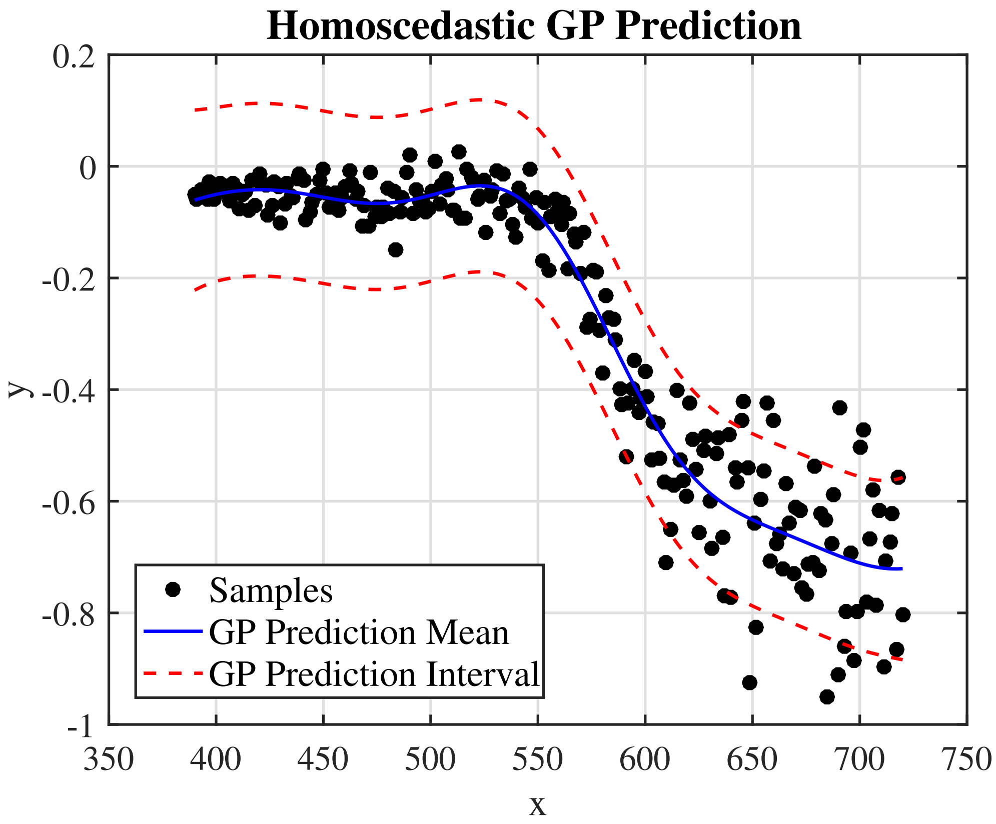


Figure I.2: Inaccuracy in homoscedastic GP's prediction in modelling heteroscedastic data

Mainly, the latter can be classified into models that do not require replications (Bae, Clark, and Forster, 2019; Bishop and Quazaz, 1996; Kersting et al., 2007; Lázaro-Gredilla and Titsias, 2011; Quadrianto et al., 2009; Wang, 2014; Zhang and Ni, 2020) and those that need replications (Ankenman, Nelson, and Staum, 2008; Binois, Gramacy, and Ludkovski, 2018). This thesis is focused on the HGP model that can learn functions with heteroscedastic noise without any need for replications, which has advantages in terms of experimental cost. The typical configuration for the HGP model primarily uses two GP models, i.e., one for modeling the function and another one for learning the varying noise function. The combination of these two GP models will then generate a joint distribution that models the heteroscedastic data.

The first attempt to obtain the numerical solution of this distribution is through Markov chain Monte Carlo (MCMC) samplings (Bishop and Quazaz, 1996; Wang, 2014). The MCMC-based approaches are often viewed as the standard for the inference distributions as these approaches can converge to the exact non-Gaussian posterior when the number of the training samples used in MCMC is high. However, the computational cost of MCMC is really high, especially in high-dimensional problems.

The alternative to reduce the HGP's computational cost is by using analytical approximations to model the heteroscedastic function distributions. For example, the variational heteroscedastic GP (VHGP) method (Lázaro-Gredilla and Titsias, 2011) uses a two-factor variation distribution to approximate the joint posterior distribution. However, this technique is rather complex, and its computational cost remains expensive. A simpler alternative is to use the most likely noise approach, introduced in the most likely noise HGP model (MLHGP) (Kersting et al., 2007; Quadrianto et al., 2009), which replaces a point estimate at its most likely level to the noise posterior that allows for an analytical predictive posterior distribution. This MLHGP model has been widely used for many practical applications due to its attractive computational efficiency (Menzer et al., 2013; Zhu et al., 2020). However, it has been shown that the MLHGP model is not guaranteed to converge as it might oscillate due to its empirical estimation of the varying noise variances (Zhang and Ni, 2020).

The improved MLHGP (IMLHGP) (ibid.) model was then proposed to overcome the numerical inaccuracy and instability of the MLHGP approach. In the IMLHGP procedure, the noise variances are estimated by utilizing the moment estimation of Bayesian residuals. They also showed that the noise variance estimation in MLHGP is biased towards a constant global noise variance, while the procedure in IMLHGP is approximately unbiased. The procedure of IMLHGP also avoids the expectation-maximization (EM)-like learning procedure as in the MLHGP, resulting in a more efficient computational cost. The computational cost to construct IMLHGP is only twice the standard GP. In their original paper, IMLHGP also showed to have better performances in predicting heteroscedastic functions than MLHGP and similar performances to that of VHGP in high training sample size cases. The IMLHGP model could then be viewed as a state-of-the-art model within the most likely noise approach HGP models.

However, in the present work, IMLHGP is found to require a high number of training samples as the model tends to fail in capturing the noise level trend when the training sample size is rather low; its predition seems to interpolate the Bayesian residuals which does not represent the true noise variances. With a low training sample size, IMLHGP is no longer gives fine prediction on the heteroscedastic function, especially for its noise level. This problem is observed to be mainly caused by the Bayesian residuals that is highly scattered when the training sample size is low, especially in the region where the noise variances are high. The highly scattered Bayesian residual value, when utilized in constructing the IMLHGP model, results in a poor and unstable predictive distribution.

The large number of samples required to train the IMLHGP model is deemed unfavorable when the cost of obtaining the samples is expensive, such as in aerospace problems. In reducing the required number of samples to train the model, while still taking advantage of the IMLHGP's computational attractivity, a modified most likely noise approach HGP model based on the IMLHGP is proposed in the present work. The proposed model, Nearest Neighbor Point Estimate HGP (NNPEHGP), follows the main idea used in MLHGP and IMLHGP by replacing a point estimate to the full noise posterior distribution. The procedure also follows the same transformation on the Bayesian residuals using the moment estimation method as with the IMLHGP method to avoid iterative learning process. The modification in this proposed NNPEHGP model is by augmenting an extra distance-based, non-parametric regression, in the IMLHGP procedure to avoid the problem encountered by IMLHGP in predicting the noise function.

To validate the performance of the proposed NNPEHGP, its predictive accuracy is compared by learning several heteroscedastic problems with various training sample sizes. The present work considers; two mathematical functions, two real-world experimental datasets, two real-world analytical problems, and a stochastic simulator. Results indicate that the proposed NNPEHGP model, contrast to the previous IMLHGP model, provides more stable predictions where it significantly outperforms in a situation when the training dataset is small and a slight improvement in the predictions when the training dataset is large.

Under the Laboratory of Aerodynamics, Department of Aerospace Engineering, Institut Teknologi Bandung, there have been several extensive researches done on the development of surrogate model and its utilization. Started by Dwianto (2015), where he developed surrogate-assited genetic algorithm for computationally expensive fluid flow optimization. Daniella (2015) and Fathurrohim (2016) used the optimization framework to assist experimental optimization of 2-D flapping plate kinematics and thrust performance of flexible caudal fin panels. For CFD-based aerodynamic optimization, Zakaria (2017a) developed mesh deformation based on RBF interpolation to automate the shape deforming procedure in the optimization process. The research is continued on developing framework for multi-objective optimization purpose (Jim et al., 2020; Palar et al., n.d.; Zuhal et al., n.d.[a],[c]; Zuhal, Palar, and Shimoyama, 2019), started with Faza (2018), which he developed a Kriging infill criterion called expected hypervolume improvement (EHVI). Rahmad (2019) also used the multi-objective optimization framework to develop a balance inverse-direct optimization to improve the performance of an airfoil. Robani (2019) continued the reseach on optimization by developing a Kriging infill criterion called Approximate Knowledge Gradient (AKG) to optimize noisy problems. In developing the surrogate model, Zuhal et al. (n.d.[b]) published a paper on Gradient enhanced universal Kriging with Polynomial Chaos Expansion, based on his master's thesis works (Zakaria, 2019). Faza (2020) then extend the ability of GP surrogate model to learn high-dimensional problems by developing a dimensionality reduction method. And recently, Izzaturahman (2020) developed Support Vector Machine (SVM) surrogate model to predict the aeroelastic flutter boundary. Several works have also been conducted in these areas of research: robust optimization (Palar et al., 2016), uncertainty quantification (Palar, Shimoyama, and Zuhal, 2020), and constrained optimization (Palar et al., 2019).

In the conducted researches under the Laboratory of Aerodynamics, Department of Aerospace Engineering, Institut Teknologi Bandung, there have not been any works focused on developing GP for heteroscedastic problems. This topic seems to be important, considering the works in the biomimetics experiment where the utilization of physical laboratory experiment is often inevitable, since the works involving complex design configurations. Fathurrohim (2016) showed that the results obtained in his fin propulsor experiment has a significant varied noise level in the observed samples, that can be seen in Fig. I.3. The biomimetics researches that have been conducted in Aerodynamics Laboratory FTMD ITB still used the

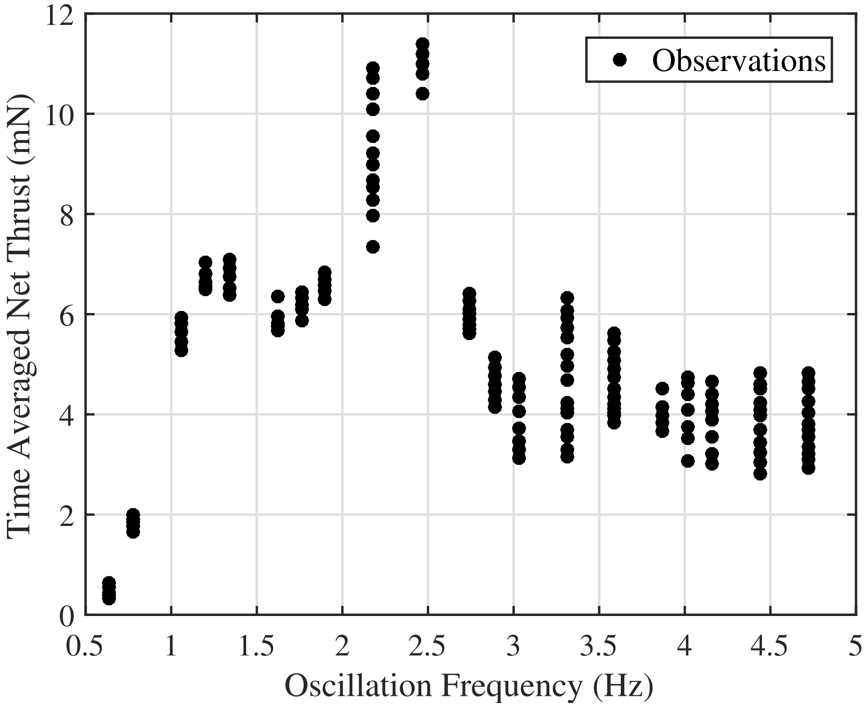


Figure I.3: Samples observed in the fin propulsor experimental data. Adapted from Fathurrohim, 2016

homoscedastic GP as the surrogate model in the design explorations (Daniella, 2015; Fathurrohim, 2016; Iqbal, 2018), that may be inappropriate for such heteroskedastic samples.

For future research regarding this kind of data, the research conducted in this thesis is expected to give a more appropriate surrogate model alternative that might lead to a more accurate prediction. For example, the varying variances of the sample could be predicted that can be utilized to investigate certain aerodynamic phenomena caused them. The experiments also could be done more efficiently by avoiding additional replications at each design possibility.

I.2 Research Problems

Problems to be answered by the research in this thesis is concerning on how to reduce the required number of samples to construct a most likely noise approach

HGP, based on the IMLHGP procedure, that can give fine predictions on the heteroscedastic functions. This research problems can be stated as a following question:

1. How to construct most likely approach HGP that requires low training samples while giving fine predictions on the heteroscedastic problems?

I.3 Research Objectives

To answer questions stated from the research problems, several objectives are defined as:

1. To develop a modified most likely noise approach HGP procedure that can learn heteroscedastic problems with a lower requirement on the number of samples.
2. To compare the performances of the proposed HGP model to other GP models.

I.4 Research Scope

To limit this thesis, the scope of work on this thesis is outlined as follows:

1. Gaussian process model

   Gaussian process model used in this thesis is the ordinary Kriging with a constant mean GP prior. This current research is also focused on using squared exponential (SE) covariance function to define the covariance matrix.

2. Type of the noises

   This current research is focused on using the most likely noise approach in developing the HGP model. This approach is designed to deal with noise that came from Gaussian distribution. Therefore, the HGP model in this work is also developed to deal with Gaussian distributed noise.

3. Computational Experiments

   Since this thesis focused on developing a HGP model, the main works is on comparing the predictive performance of the models to the true values. When the true values is unknown, the reference values that is expected to have similar values to the true one can be obtained by evaluating numerous realizations at each input location. Therefore, the problems to compare the performances of the model in this thesis is chosen to have a cheap computational cost to evaluate. Since the proposed NNPEHGP model uses the concept of most likely noise approach HGP, this model is only compared to IMLHGP and standard homoscedastic GP. IMLHGP is chosen as the benchmark model for the most likely noise approach HGP while standard homoscedastic GP is chosen to represent the basic approach of GP in learning noisy problems.

I.5 Thesis Outline

This thesis is consisted of 5 chapters. The outline and brief explanation is given as follows:

1. Chapter I: Introduction

   This chapter describes the historical background of the research, the problems, the objectives, scope of work and the outline of the thesis.

2. Chapter II: Fundamental Theories

   This chapter explains the fundamental concepts of Gaussian process (GP) surrogate model and summarizes the foundation of heteroscedastic GP (HGP). The chapter is continued by discussing the most likely noise approach HGP model and the IMLHGP as the benchmark model for this study. This chapter is closed by the explanation of the proposed HGP model that utilizes nearest neighbor point estimates.

3. Chapter III: Research Methodology

   This chapter contains the setup for the computational experiments and summarizes the test case problems.

4. Chapter IV: Surrogate Models Performance

   This chapter discussed the performance of the models in learning each test case problems.

5. Chapter V: Conclusion and Future Works

   This chapter gives conclusion of this current research and recommendations on future researches.

Chapter II

Fundamental Theories

This chapter explains the fundamental concepts of Gaussian process (GP) surrogate model in Section II.1 and summarizes the foundation of heteroscedastic GP (HGP) in Section II.2. The chapter is continued by discussing the most likely noise approach HGP model in Section II.2.1 and the improved most likely HGP (IMLHGP) as the benchmark model for this study (Section II.2.2). This chapter is closed by the explanation of the proposed HGP model that utilizes nearest neighbor point estimates in Section II.3.

II.1 Foundation of Gaussian Process

Gaussian process (GP), as a surrogate model, approximates an unknown black-box function $y = f(\boldsymbol{x})$ based on finite number of observed samples $D = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, n\}$, where $\boldsymbol{x} \in \mathbb{R}^m$ denotes the input vector with dimension m and n is the number of samples to train the model (training samples). The unknown function can be described as,

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i \text{ with } \varepsilon_i \sim \mathcal{N}(0,\, g^2(\boldsymbol{x}_i)) \tag{II.1}$$

where the observation here is assumed to be independent and normally distributed, with the mean of each observation is located at the true function $f(\boldsymbol{x}_i)$ and its variance is described with the noise function $g^2(\boldsymbol{x}_i)$. The samples' value, the true function value, and the noise standard deviation at training inputs $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ here can be collected into a vector as $\boldsymbol{y} = (y_1, \ldots, y_n)$, $\boldsymbol{f} = (f_1, \ldots, f_n)$, and $\boldsymbol{g} = (g_1, \ldots, g_n)$, respectively.

This unknown black-box function can be treated as a mapping of any value y, of interest, from a set of defined input parameters $\boldsymbol{x}$. When the samples are acquired from deterministic simulations, such as a typical computer simulation, the noise function g can be considered to be zero and the surrogate model would focus on learning the true function f only. The modelling will becoming more complex when noise corrupts the functions since the noise function g has to be considered in appropriately learning the samples. The noise level on the samples that is described with the noise function g could be constant (homoscedastic) or varied across the input spaces (heteroscedastic).

In this work, the surrogate model will be focus only on the GP surrogate model as GP can learn, not only the true (mean) function but also the distribution over observation at each input that describes the noise variances. The GP's capability is clearly beneficial in engineering design exploration, where the robustness of each design possibility not only its expected performance is often needed to be learned. For example, in the context of robust optimization, we want to have a design that has a high performance with low variability in its uncertain operational conditions. This kind of application need to have a surrogate model that can accurately predict the mean and its noise level.

This section will be continued by explaining the fundamentals of GP surrogate model, on how to construct the model and how the model gives predictions over the samples.

II.1.1 GP Prior

GP is basically a form of non-parametric regression, where one does not have to necessarily know prior insights about the data, as opposed to a parametric regression model. This non-parametric form of GP avoids the model to be biased towards certain characteristics of the function.

GP constructs the surrogate model by treating the observed samples in a function-space framework. Where in this framework, the procedure is done by assigning probabilities to each of the infinite number of possible functions to model the unknown black-box function. The function with a higher probability is considered to be more likely as the approximate mapping model to the samples. The high flexibility of the GP model in learning the black-box functions is seen by working in this function-space framework. However, the procedure to learn the samples in this function-space framework without any prior assumption will largely increase the computational cost, as each of the infinite possible functions needs to be constructed.

In relaxing the computational process while working in this function-space framework, GP assumes the probability of these infinite set of functions to be governed by a stochastic process that is Gaussian. GP models the n number of samples to adopt a multivariate normal (MVN) distribution that is fully defined by a $(n \times 1)$ mean vector $\boldsymbol{\mu}$ and a $(n \times n)$ covariance matrix $\boldsymbol{K}$. This assumption, that can also be thought as the GP prior, is placed over the observed samples in constructing the surrogate model, such as

$$p(\boldsymbol{y}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \tag{II.2}$$

In this work, the values in the mean vector $\boldsymbol{\mu}$ are considered to be constant (i.e., ordinary Kriging) and the (i, j) entry of the covariance matrix $\boldsymbol{K}$ is mainly calculated between each of the observed samples as,

$$cov(y_i, y_j) = c(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{II.3}$$

where c here is the covariance function, that will be explained in Section. II.1.2. This covariance matrix $\boldsymbol{K}$ essentially describes the correlation between each of the observed samples, that is considered to be random variables in defining the MVN distribution.

Training a GP surrogate model based on a n number of observations data is essentially governed by the process of obtaining the best parameters to define the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{K}$ in modelling the unknown function. These parameters, often called as the GP hyperparameters, will be explained in the next section.

II.1.2 GP Hyperparameters

In constructing the surrogate model, GP has several hyperparameters to define the MVN distribution conditioned on the data. This GP hyperparameters are described as the parameters to define the mean vector and covariance matrix. Since this thesis only focused on using the constant mean vector, the explanation in this section will be focused on the hyperparameters to define the covariance matrix and the procedure to get the best hyperparameters in learning the unknown relation of the samples.

Covariance Function Parameters

In conditioning the MVN distribution on the data, the first procedure is to determine how the samples in the data relates with one another. This relationship can be determined by the covariance function, that mainly computes the quantity based on the distance of the data. As mentioned in the Section II.1.1, this covariance function is used to compute each element in the covariance matrix for the GP prior. By assuming that the data is jointly Gaussian distributed, this covariance matrix requires a characteristic of a MVN distribution covariance matrix, that must be a semi positive-definite one. Therefore, not all distance-based functions of an input pairs $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be used as a valid covariance function.

In this thesis works, the popular squared-exponential (SE) covariance function is chosen as the covariance function in defining the covariance matrix. The SE covariance function itself can be read as,

$$c(\boldsymbol{h}, \sigma_f^2, \boldsymbol{l}) = \sigma_f^2 \prod_{k=1}^{m} \exp\left[-0.5\left(\frac{h_k}{l_k}\right)^2\right] \tag{II.4}$$

where $\boldsymbol{h}$, σ_f^2 and $\boldsymbol{l}$ are the distance metric between two points, signal variance and the length scale of the covariance function, respectively. The signal variance here explains the variability of the GP prior functions. The distance metric chosen here is the Euclidean distance between two input locations $(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The Euclidean distance at each dimension of the inputs can be calculated as

$$\boldsymbol{h}_k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i^k - \boldsymbol{x}_j^k)^2} \tag{II.5}$$

where k here denotes the quantity at each dimension of the input locations.

The length scale in the covariance function represents the influence of a sample on other samples in the data. The values are defined in a $(m \times 1)$ vector, with each element corresponds to the length scale value in each dimension of the input. A larger value of this length scale parameter shows a higher influence between each sample to other samples in the corresponding dimension.

Noise Parameter

When the covariance matrix is purely defined as the values calculated from the covariance function, GP will yield an interpolating prediction to the samples. This would not be a problem if the observed samples are obtained from a deterministic experiment, such as typical computer simulation, since each observed sample is the only possible value to a corresponding set of input parameters. However, when the samples are obtained from a noisy experiment, such as physical experiments or stochastic simulators, the observed samples could not be taken as the only possible values; an interpolating prediction is very likely to be wrong in these kind of noisy samples.

When dealing with noisy samples, the covariance matrix has to be modified. This modification can be done by treating the covariance matrix as,

$$\mathbf{K} = \mathbf{C} + \mathbf{S} \tag{II.6}$$

where the $\mathbf{C}$ matrix contains the values calculated from the covariance function and the $\mathbf{S}$ matrix is the matrix in representing the uncertainties to the samples, or later will be referred as the noise matrix. The noise matrix $\mathbf{S}$ is a diagonal matrix, in which the (i, i) value represents the noise variance g^2 at each training input $\boldsymbol{x}_i$.

A simple way to assign the values in this noise matrix is by taking a constant parameter to all the diagonal elements in the matrix, assuming the noise variances to be constant at each of the training inputs. This is the approach of the standard homoscedastic GP, which fills the diagonal noise matrix as,

$$\mathbf{S} = \sigma_n^2 \mathbf{I} \tag{II.7}$$

where σ_n^2 is the constant noise parameter and $\mathbf{I}$ is a $(n \times n)$ identity matrix. This approach allows the GP model to steer away from interpolating the noisy samples anymore.

Maximum Likelihood Estimation

As the GP mainly learn the function by conditioning a MVN distribution on the observed samples, the expected performance of the GP in modelling the samples is also calculated by this idea. As mentioned earlier, training a GP model is primarily focused on obtaining the best hyperparameters in defining the distributions of data. Therefore, a quantitative measurement on estimating the quality of the model with a set of hyperparameters need is required. To determine the best hyperparameters, a common procedure is the maximum likelihood estimation (MLE). MLE is a procedure to maximize the log marginal likelihood of the data, conditioned on the MVN distribution defined by a set of GP hyperparameters. Describing the hyperparameters in a vector, i.e., $\theta_y = \{\boldsymbol{l}, \sigma_f^2, \sigma_n^2, \boldsymbol{\mu}\}$, the log marginal likelihood of the data can be described as

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \theta_y) = -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})^T(\mathbf{C} + \mathbf{S})^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) - \frac{1}{2}\log|\mathbf{C} + \mathbf{S}| - \frac{n}{2}\log(2\pi) \tag{II.8}$$

The best hyperparameters is estimated by maximizing this log marginal likelihood equation, that is essentially an optimization process with the possible set of GP hyperparameters as the inputs. This optimization is usually conducted by utilizing an extra optimizer module, that could be stochastic or gradient-based. In this thesis, the popular CMAES optimization algorithm (Hansen, Müller, and Koumoutsakos, 2003) is used to estimate the set of hyperparameters that maximizes the log-marginal likelihood over the observed samples.

II.1.3 GP Posterior

With the GP prior already defined in Eq. II.2, the observed samples $\boldsymbol{y}$ and the values f_* at an arbitrary test location are jointly Gaussian distributed as

$$\begin{bmatrix} \boldsymbol{y} \\ f_* \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu \end{bmatrix}, \begin{bmatrix} \mathbf{C}+\mathbf{S} & \mathbf{c}_* \\ \mathbf{c}_* & \mathrm{c}_{**} \end{bmatrix}\right) \tag{II.9}$$

where $\mathbf{c}_*$ is the covariance vector between training inputs $\boldsymbol{X}$ and test input $\boldsymbol{x}_*$ calculated by $c(\boldsymbol{x}_*, \boldsymbol{X})$ and c_{**} is the prior variance at the test input $\boldsymbol{x}_*$ calculated by $c(\boldsymbol{x}_*, \boldsymbol{x}_*)$. This conditional identity of multivariate Gaussian distribution (Eq. II.9) can be used to calculate the posterior distribution of the function value f_* at a test input $\boldsymbol{x}_*$, with a given set of hyperparameters, that can be read as

$$p(f_*|\boldsymbol{x}_*, \theta_y, \boldsymbol{g}, \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\mu_{f_*}, \sigma^2_{f_*}), \text{ where} \tag{II.10}$$

$$\mu_{f_*} = \mu + \mathbf{c}_*(\mathbf{C}+\mathbf{S})^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{II.11}$$

$$\sigma^2_{f_*} = \mathrm{c}_{**} - \mathbf{c}_*(\mathbf{C}+\mathbf{S})^{-1}\mathbf{c}_* \tag{II.12}$$

The posterior distribution over test output y_* can be obtained by adding the noise variance at test output $g^2(\boldsymbol{x}_*)$ to the posterior variance of the function value $\sigma^2_{f_*}$. This relation came from the function definition in Eq. II.1. The predictive posterior distribution over test output y_* then can be stated as

$$p(y_*|\boldsymbol{x}_*, \theta_f, \boldsymbol{g}, \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\mu_{y_*}, \sigma^2_{y_*}), \text{ where} \tag{II.13}$$

$$\mu_{y_*} = \mu + \mathbf{c}_*(\mathbf{C}+\mathbf{S})^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{II.14}$$

$$\begin{aligned} \sigma^2_{y_*} &= \sigma^2_{f_*} + g^2(x_*) \\ &= \mathrm{c}_{**} - \mathbf{c}_*(\mathbf{C}+\mathbf{S})^{-1}\mathbf{c}_* + g^2(x_*) \end{aligned} \tag{II.15}$$

When the standard homoscedastic GP is used to model the observed data, the noise variance term at a test input $\boldsymbol{x}_*$ is assumed to have the same values as the noise parameter σ_n^2 at each training input. Which implies that the samples are assumed to have a constant noise level. The assumption of this constant noise variance is certainly not appropriate when the function has varying noise variances across its input spaces. The heteroscedastic GP (HGP) model is then developed to deal with the samples that has an input-varying noise level. The next section will be focused in explaining the foundation of this HGP model.

II.2 Heteroscedastic GP

A typical approach, to model a heteroscedastic problem with the use of GP, is to construct two GP models, where the first GP is used to learn the true function f and the second GP is used to model the noise level g. These two GP models are constructed with a separate set of hyperparameters.

The noise level is usually transformed prior to its modeling, to make sure the values are positive in the input spaces. For example, by taking the logarithmic values of the noise variances, such as $z_i = \log(g_i)$. The second GP is then constructed to model this transformed noise level, parameterized by a separate set of hyperparameters $\boldsymbol{\theta}_z$.

The predictive distribution over test output y_* at a test input $\boldsymbol{x}_*$ could be calculated by this following integration

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_f, \boldsymbol{\theta}_z, \boldsymbol{y}, \boldsymbol{X}) = \int\int p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_f, \boldsymbol{z}, z_*, \boldsymbol{y}, \boldsymbol{X}) p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \boldsymbol{\theta}_z, \boldsymbol{y}, \boldsymbol{X}) d\boldsymbol{z} dz_* \tag{II.16}$$

where $\boldsymbol{z} = (z_1, \ldots, z_n)^T$ are the transformed noise level at training inputs $\boldsymbol{X}$ and z_* is the transformed noise level at an arbitrary input $\boldsymbol{x}_*$. Notice that when the full distribution of the noise level $p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \boldsymbol{\theta}_z, \boldsymbol{y}, \boldsymbol{X})$ is considered, this integral could not be solved analytically. The full predictive distribution at each input could be estimated, for example, by utilizing the Markov Chain Monte Carlo (MCMC) procedure that considers each possible value from each noise level distribution. However, this procedure will leads to an excessive amount of computational cost, especially in the high dimensional input parameters. To simplify the HGP model when the actual noise level is unknown, one approach that can be used is by estimating a value of the noise level at each input, rather than its full distribution. This is the treatment of the most likely noise approach HGP that will be explained in the following section.

II.2.1 Most Likely Noise Approach HGP

The central idea of the most likely noise noise approach HGP, first introduced as the Most Likely HGP (MLHGP) (Kersting et al., 2007), is to replace the full posterior distribution of the noise variance with a point estimate at the most likely value. The use of a point estimate allows the predictive posterior distribution over test output to be computed analytically. To that end, MLHGP approximates the noise posterior $p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \theta_z, \boldsymbol{y}, \boldsymbol{X})$ by a Dirac delta function, reads as

$$p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \boldsymbol{\theta_z}, \boldsymbol{y}, \boldsymbol{X}) = \begin{cases} 1, & \text{if } \boldsymbol{z} = \tilde{\boldsymbol{z}} \text{ and } z_* = \tilde{z}_* \\ 0, & \text{otherwise} \end{cases} \tag{II.17}$$

where $\tilde{\boldsymbol{z}}$ and $\tilde{z}_*$ is the most likely log noise level at training inputs $\boldsymbol{X}$ and test input $\boldsymbol{x}_*$, respectively.

The predictive posterior distribution over test output then can be approximated as

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_f, \boldsymbol{\theta}_z, \boldsymbol{y}, \boldsymbol{X}) \approx p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_f, \tilde{\boldsymbol{z}}, \tilde{z}_*, \boldsymbol{y}, \boldsymbol{X}) \tag{II.18}$$

where the most likely noise level $\tilde{\boldsymbol{z}}$ and $\tilde{z}_*$ is at the mode of its noise posterior, reads as

$$(\tilde{\boldsymbol{z}}, \tilde{z}_*) = \underset{(\boldsymbol{z}, z_*)}{\arg\max} \log p(\boldsymbol{z}, z_*|\boldsymbol{x}_*, \boldsymbol{\theta}_z, \boldsymbol{y}, \boldsymbol{X}) \tag{II.19}$$

Notice that since the noise level is modeled by another GP, its posterior distribution is also Gaussian and the mode of each noise posterior is located at the noise posterior mean. The most likely noise level is then given as

$$(\tilde{\boldsymbol{z}}, \tilde{z}_*) = (\boldsymbol{\mu_z}, \mu_{z_*}) \tag{II.20}$$

where $\boldsymbol{\mu_z}$ and μ_{z_*} are the posterior means of transformed noise levels at training inputs $\boldsymbol{X}$ and test input $\boldsymbol{x}_*$, respectively. The predictive distribution over test output y_* then can be calculated by

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_f, \boldsymbol{\theta}_z, \boldsymbol{y}, \boldsymbol{X}) &\approx p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_f, \tilde{\boldsymbol{z}}, \tilde{z}_*, \boldsymbol{y}, \boldsymbol{X}) \\ &= p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_f, \boldsymbol{\mu_z}, \mu_{z_*}, \boldsymbol{y}, \boldsymbol{X}) \end{aligned} \tag{II.21}$$

It can be seen that the predictive posterior distribution over test output y_* is Gaussian distributed, with its mean and variance given by Eq. II.13, by inverse transforming the z-values $\boldsymbol{z}$ to its noise level $\boldsymbol{g}$.

The noise level at training inputs have to be estimated first before the second GP model can be used to learn and predict the values along the input spaces. The empirical procedure to estimate this noise level is what differentiate between the MLHGP, improved most likely HGP (IMLHGP), and proposed model in this thesis works, i.e., nearest neighbor point estimates HGP (NNPEHGP). In MLHGP, the empirical estimate of the noise level is given as

$$g_i^2 = \frac{1}{s} \sum_{j=1}^{s} 0.5(y_i - y_i^j)^2 \tag{II.22}$$

where y_i^j are samples generated using the predictive posterior distribution of the training output y_i with s number of samples (the distribution is Gaussian with mean μ_{y_i} and variance $\sigma_{y_i}^2$). Due to its Gaussian nature, Eq. II.22 can be further simplified into

$$g_i^2 = \frac{1}{2} \left[(y_i - \mu_{y_i})^2 + \sigma_{y_i}^2 \right] \tag{II.23}$$

Using this simplification, the noise variances can be estimated without generating any samples from the predictive distribution, that can reduce the computational cost and enhance its numerical stability. A new training dataset, $z_i = \log\{\frac{1}{2}((y_i - \mu_{y_i})^2 + \sigma_{y_i}^2)\}$ at training inputs $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}^T$, can then be used to train another GP for estimating the most likely noise variance, as $\tilde{g}_i^2 = \exp(\mu_{z_i})$.

The algorithm of MLHGP starts by fitting a standard homoscedastic GP over the observed samples and estimate the noise variances at training inputs using Eq. II.23 with the already obtained posterior distribution. The next step is to train another standard homoscedastic GP model on the transformed noise variance (its logarithmic values) to obtain the most likely noise variances at each input point. Another GP is then trained using these most likely noise variances to obtain the predictive posterior distribution over test outputs by using Eq. II.14 and Eq. II.15. This procedure is then repeated until the convergence is reached. Although MLHGP is much simpler and also more computationally efficient than MCMC-based methods, such an approach is not guaranteed since the estimated noise variance is biased toward a constant noise variance in the input space (Zhang and Ni, 2020). In improving the performance of MLHGP, the IMLHGP is proposed to avoid the biased assumption in MLHGP procedure.

II.2.2 Improved Most Likely HGP

To overcome the biased nature of MLHGP, IMLHGP offers an improvement by using an approximately unbiased approach based on the moment estimation of Bayesian residuals to estimate the noise variances. The Bayesian residuals r_i are defined as the difference between the evaluated outputs y_i and the posterior means μ_{y_i} at each training input $\boldsymbol{x}_i$, that is,

$$r_i = y_i - \mu_{y_i} \tag{II.24}$$

In contrast to the classical residuals which are defined as the difference between the observed outputs y_i and the point estimates $\hat{y}_i$.

By a simple manipulation, the Bayesian residuals r_i can also be rewritten as

$$\begin{aligned} r_i &= y_i - \mu_{y_i} \\ &= y_i - f_i + f_i - \mu_{y_i} \\ &= \varepsilon_i + (f_i - \mu_{y_i}), \end{aligned} \tag{II.25}$$

where $\varepsilon_i = (y_i - f_i)$ is the observation error and $(f_i - \mu_{y_i})$ is the modeling error. As explained in Eq. II.1, the observation error ε_i is considered as a random variable that is normally distributed as $\mathcal{N}(0,\, g_i^2)$ while the modelling error $(f_i - \mu_{y_i})$ is a deterministic variable. The Bayesian residuals r_i then are also random variables, which are distributed as

$$r_i \sim \mathcal{N}(\mu_{r_i},\, \sigma_{r_i}^2) \tag{II.26}$$

where its mean μ_{r_i} is the modelling error $(f_i - \mu_{y_i})$ and its variances $\sigma_{r_i}^2$ are the variances of the observation error (noise variances) g_i^2. By training a GP model to the observed outputs $\boldsymbol{y}$, the Bayesian residual r_i at each training input $\boldsymbol{x}_i$ and the posterior means of training outputs μ_{y_i} can be simply calculated. While the true function value f_i and the noise variances g_i^2 still need to be estimated.

In the IMLHGP approach, the Bayesian residuals r_i are utilized to estimate the varying noise variances. IMLHGP uses a common practice for parameter estimation; namely, the method of moments (Hogg, Tanis, and Zimmerman, 2010), to provide an unbiased estimate for the noise variances. Its local noise variances can be derived from various available statistical moments of the Bayesian residuals r_i, such as raw or central moments and raw or central absolute moments. In IMLHGP, the Bayesian residuals' raw absolute moments are used as it contains information of the varying noise variances.

The v-th raw absolute moment of the Bayesian residual r_i at each training input $\boldsymbol{x}_i$ is given by

$$\begin{aligned} \mathrm{E}\{|r_i|^v\} &= \frac{\sigma_{r_i}^2}{s(v)} \\ &= \frac{g_i^v}{s(v)} \end{aligned} \tag{II.27}$$

where $s(v)$ is the correction factor that depends on the moment order v (Winkelbauer, 2012) , given by

$$s(v) = \frac{\sqrt{\pi}}{2^{v/2}\Gamma((v+1)/2)\psi(-v/2, 1/2; -\mu_{r_i}^2/2\sigma_{r_i}^2)} \tag{II.28}$$

where $\psi(\cdot)$ is the Kummer's confluent hypergeometric function and $\Gamma(\cdot)$ is the gamma function. Based on the Eq. II.27, the local noise variance g_i can be obtained by

$$g_i^v = \mathrm{E}\{|r_i|^v\}s(v) \tag{II.29}$$

This noise variances estimation procedure is suggested by the authors of IMLHGP as an unbiased estimate of the local noise variance g_i^v at each training input $\boldsymbol{x}_i$.

To learn the true function, when the GP model for the y-process is well defined, the modeling error $(f_i - \mu_{y_i})$ can be neglected as its values are close to zero. Thus, the Kummer's confluent hypergeometric function value can be taken as $\psi(-v/2, 1/2; 0) = 1$. The raw absolute moment of the Bayesian residual then degenerates to its central absolute moment, resulting in the following approximation of the correction factor $s(v)$

$$s(v) \approx \frac{\sqrt{\pi}}{2^{v/2}\Gamma((v+1)/2)} \tag{II.30}$$

As can be observed from Eq. II.29, the parameter v controls the order of the Bayesian residual moment and the order of noise level as well. When v equals to one, the learned values is the noise standard deviation g and the noise variance is learned when v equals to two.

The first absolute moment of the Bayesian residual ($v = 1$) then can be used to estimate an approximately unbiased noise standard deviation g_i at each training input $\boldsymbol{x}_i$ as

$$\begin{aligned} g_i &= \mathrm{E}\{|r_i|^1\}s(1) \\ &\approx \sqrt{\pi/2}\mathrm{E}\{|r_i|^1\} \end{aligned} \tag{II.31}$$

Similarly, the second absolute moment of the Bayesian residual ($v = 2$) can be used to estimate the noise variance g_i^2 at each training input $\boldsymbol{x}_i$, that is

$$\begin{aligned} g_i^2 &= \mathrm{E}\{|r_i|^2\}s(2) \\ &\approx \mathrm{E}\{|r_i|^2\} \end{aligned} \tag{II.32}$$

With the Bayesian residuals r_i already available from the first GP, the second GP then can be constructed with $z_i = |r_i|^v$ at training inputs $\boldsymbol{X}$ as the training samples. The expectation of the predictive posterior distributions from this second GP is then used to estimate the most likely noise levels, reads as

$$\tilde{g}_i^v = \mu_{z_i} s(v) \tag{II.33}$$

at training point $\boldsymbol{x}_i$ and

$$\tilde{g}_*^v = \mu_{z_*} s(v) \tag{II.34}$$

at test point $\boldsymbol{x}_*$, where μ_{z_i} and μ_{z_*} are the posterior means of the second GP at training inputs $\boldsymbol{x}_i$ and the test input $\boldsymbol{x}_*$, respectively. In the original paper of IMLHGP (Zhang and Ni, 2020), their results indicate that there are no significant differences between using either $v = 1$ or $v = 2$. In light of these results, $v = 2$

is chosen to be used in this study since the procedure will estimate the noise variances that can be added directly to the diagonal of the noise matrix in the GP model.

Since the prediction of the mean values in the second GP might be negative, the most likely noise levels need to be refined as

$$\tilde{g}_i^v = \max(0, \mu_{z_i} s(v)) \tag{II.35}$$

for the values at each training input $\boldsymbol{x}_i$ and

$$\tilde{g}_*^v = \max(0, \mu_{z_*} s(v)) \tag{II.36}$$

for the value at an arbitrary test input $\boldsymbol{x}_*$. This step is done to make sure the estimated noise levels are not negative. The utilization of a constant-mean in GP prior is also used in this study rather than zero-mean to make the negative prediction values appears less frequently. The full procedure to constructs the IMLHGP model is explained in Section II.2.2.

Constructing IMLHGP

1. Train a standard homoscedastic GP GP_1 on the training samples $D = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, n\}$ and estimate the posterior distribution over training outputs as
$$y_i | \boldsymbol{x}_i, \boldsymbol{\theta}_y, D \sim \mathcal{N}(\mu_{y_i}, \sigma_{y_i}^2) \tag{II.37}$$

2. Calculate the Bayesian residuals $r_i = y_i - \mu_{y_i}$ and build a new training dataset D' for the second GP, $D' = \{(\boldsymbol{x}_i, z_i) | i = 1, \ldots, n\}$ with $z_i = |r_i|^v$.

3. Train a second standard homoscedastic GP GP_2 on the new dataset D' and estimate the posterior mean over the v-th absolute moment of Bayesian residuals as
$$z_i | \boldsymbol{x}_i, \boldsymbol{\theta}_z, D' \sim \mathcal{N}(\mu_{z_i}, \sigma_{z_i}^2) \tag{II.38}$$

4. Update the most likely noise levels as

$$\tilde{g}_i^v = \max(0, \mu_{z_i} s(v)) \tag{II.39}$$

with $s(v) \approx \sqrt{\pi}/[2^{v/2}\Gamma((v+1)/2)]$

5. Make prediction on future observations with the estimated noise levels as

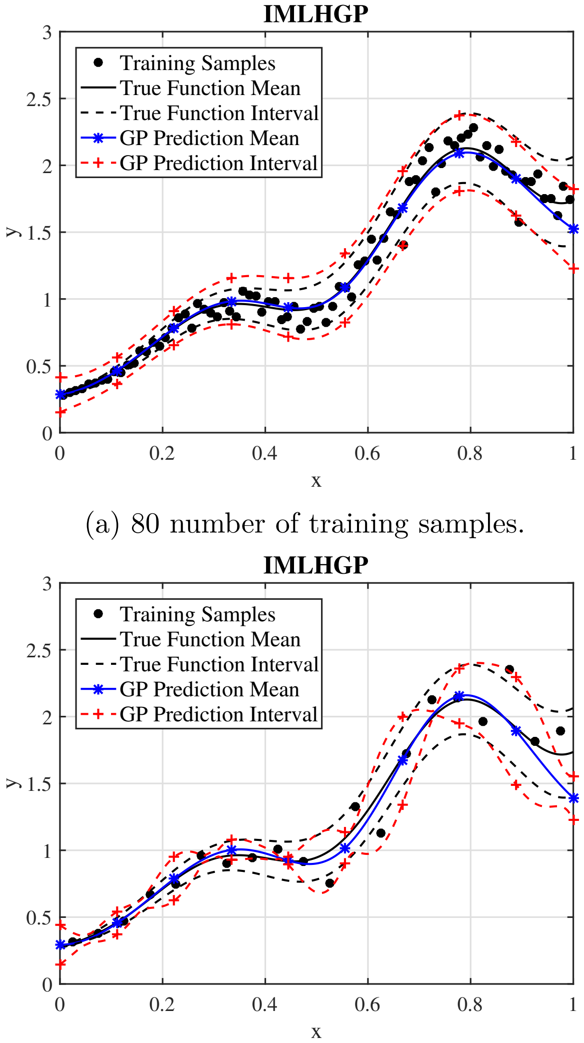
$$y_* | \boldsymbol{x}_*, \boldsymbol{\theta}_y, \tilde{\boldsymbol{g}}, \tilde{g}_*, D \sim \mathcal{N}(\mu_{y*}, \sigma_{y*}^2) \tag{II.40}$$

II.3 Nearest Neighbor Point Estimates HGP

Referring to Zhang and Ni (2020), IMLHGP demonstrates a much better prediction of the noise standard deviation and the full distributions compared to MLHGP under the condition of high training sample size. However, when a stochastic simulator/physical experiment to acquire the sample is already expensive, this high number of training samples might not be available, or at least not desirable. Regarding this, it is observed that IMLHGP is no longer yielding good predictions, especially on the noise lovel, when the size of training samples is rather low, as illustrated in Fig. II.1. As shown in Fig. II.1b, IMLHGP fails to predict the true noise level trend in the low training sample as its prediction seems to wrongly interpolate the Bayesian residuals, unlike its fine prediction when the number of training samples is high (Fig. II.1a).

This problem encountered by IMLHGP in a low training sample size is somewhat reasonable. In the case of unlucky sampling, the Bayesian residuals may often appear to approach zero, even in the region where the noise variances are high. There will not be a significant problem if the number of training samples is high since dispersed observations are available that can be used to describe the noise level trend. However, when the number of training samples is low, the Bayesian residuals are very likely to be scattered, and it becomes really difficult for GP to model this trend properly. In the worst case, GP interpolates the scattered Bayesian residuals and yield a very poor prediction of the noise variance.

The visualization of this problem is presented in Fig. II.2. It can be observed



(a) 80 number of training samples.

(b) 20 number of training samples.

Figure II.1: An example of IMLHGP predictions in predicting a one-dimensional heteroscedastic function. The interval shows 95% error variation of the true function and the GP, that represents the noise level. It can be seen that IMLHGP predicts the noise level well with 80 number of training samples. However, there is a high risk of unstable predictions in the low training sample size (in this case, 20 number of training samples)

.

that the Bayesian residuals are really scattered, especially in the high variance region (right boundary region). By the assumption of the samples will be more frequently appear in its mean (the true function), when the GP model on the y-process is well defined, this kind of samples will have the Bayesian residual

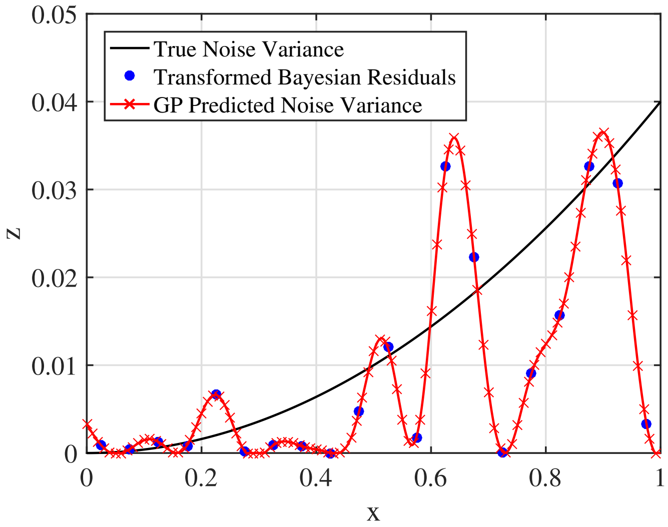


Figure II.2: Visualization of IMLHGP in learning the noise variances based on the Bayesian residuals.

value to be near zero. Even though GP has high flexibility in modelling the data, this highly scattered values still would lead to burdensome condition for GP in capturing the actual trend. This problem is then needed to avoid in efforts to improve the performance of the model when the available samples to train the model is quite low.

In overcoming this problem, Nearest Neighbor Point Estimates HGP (NNPEHGP) is proposed in this thesis by applying certain modifications to the procedure of IMLHGP. The main goal of the proposed modification is to maintain fine prediction on the heteroscedastic function in the low training sample size; reducing the required number of samples to train the model. This proposed modification in NNPEHGP procedure will be thoroughly explained in the next section.

II.3.1 Noise Level Estimation Smoothing Procedure

To maintain a good prediction on the noise level, the procedure of IMLHP is relaxed by using an extra distance-based non-parametric regression to smooth out the Bayesian residuals calculated at training inputs. The main motivation comes from the observation which shows that IMLHGP models are not capable of predicting the noise level well when the training sample size is low. It is worth noting that the second GP learns and models the Bayesian residual by a weighted average procedure. When replications are available, methods such as stochastic Kriging and practical HGP model the noise level by averaging the residual values from multiple replications at each point.

At one input point, assuming that the noise levels at neighboring samples are similar and correlated, the noise level can be estimated by averaging the Bayesian residuals of that particular point to its nearest neighbors. In this proposed NNPE-HGP, the location of nearest neighbors is considered by using two parameters: nearest number of neighbors that is defined by k and maximum distance to the nearest neighbors that is defined by d. The distance-based regression is proposed to use to perform the averaging such that the Bayesian residuals across the input spaces are smoothed out

By defining number of nearest neighbors with k, the procedure will be similar to k-nearest neighbor (kNN) regression in averaging the Bayesian residuals. The kNN regression is applied to transform the v-th absolute moment of Bayesian residual values $z_i = |r_i|^v$ from Eq. II.24 at a training input $\boldsymbol{x_i}$, such that

$$\hat{z}_i = \frac{1}{k} \sum_{\boldsymbol{x}_i \in N_k} z_i \tag{II.41}$$

where N_k is the set consisting the k nearest neighbors of a training input $\boldsymbol{x}_i$, in which the Euclidean distance is used to measure the closeness. In other words, by using Eq. II.41, the procedure finds the k observations closest to $\boldsymbol{x}_i$ for the v-th Bayesian residuals to be averaged. The value of k should be inputted by the user, in this thesis, $k = 0.2n$ and $k = 0.4n$ are used as the preliminary study to generate the results.

The second configuration for NNPEHGP is by using parameter maximum distance d to define the nearest neighbors of a training input $\boldsymbol{x}_i$. The procedure is rather similar to the kNN regression, the main difference is only on choosing the nearest neighbors. The smoothing procedure in this configuration can be read as

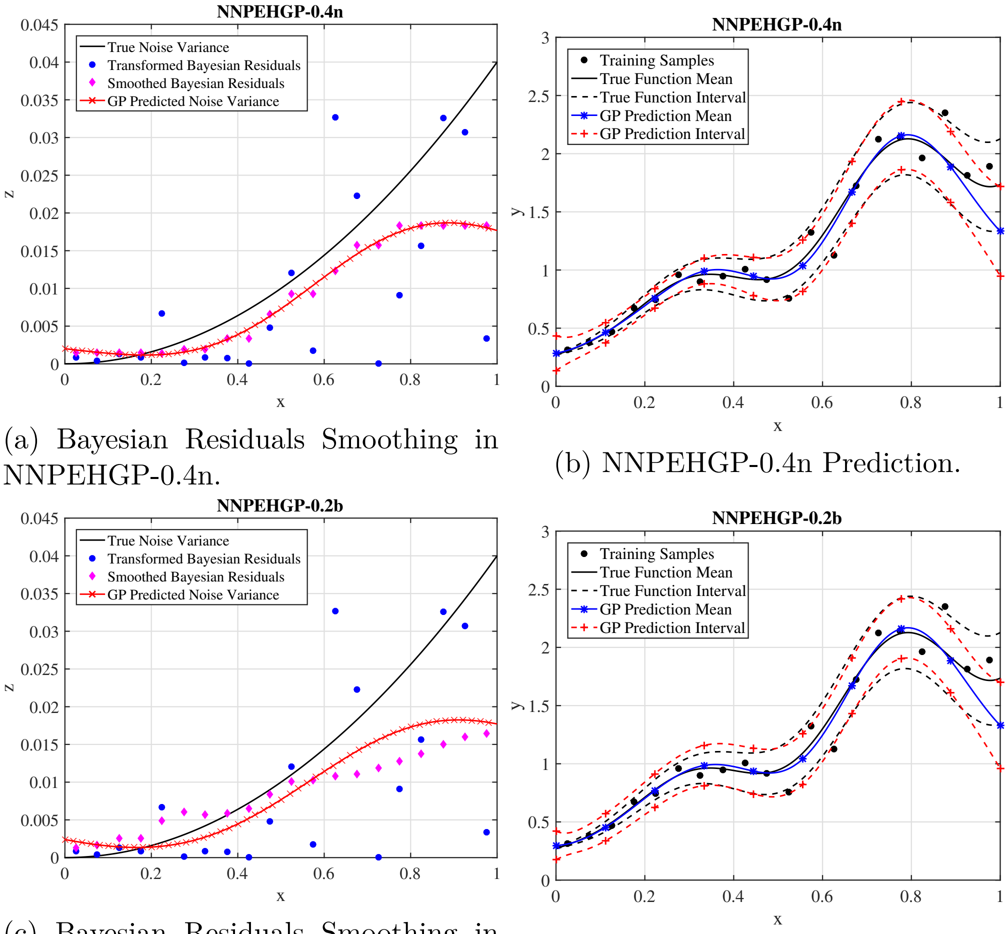
$$\hat{z}_i = \frac{1}{k_d} \sum_{\boldsymbol{x}_i \in N_d} z_i \tag{II.42}$$

where N_d is the set of training inputs consisting k_d nearest neigbors of a training input $\boldsymbol{x}_i$, that has lower distances than the defined maximum distance d. The distances is also measured by using Euclidean distance. Such as previous configuration, the value of d should be inputted by the user, in which $d = 0.2b$ and $d = 0.4b$ are used in this work, where b is the range of the input spaces.

The procedure is named as NNPEHGP to emphasize the use of nearest neighbor regression as one key component of the method. By smoothing out the Bayesian residual values at training inputs via distance-based regression and then use $\hat{z}_i$ in place of z_i to train the GP model, it is then expected that it can reduce the highly scattered Bayesian residuals in low training sample size. Hence, this might leads to a greatly suppressed the wrongly noise level predictions that appeared in IMLHGP procedure.

Fig. II.3 shows an example of the application of NNPEHGP to the similar sampling set used in Fig. II.1b. Here, the two configuration for NNPEHGP is presented, NNPEHGP with parameter k is presented in Figs. II.3a and II.3b while NNPEHGP with parameter d is presented in II.3c and II.3d. $k = 0.4n$ and $d = 0.2b$ are the parameters chosen for NNPEHGP in this example.

It can be clearly observed in Figs. II.3a and II.3c that both configurations of the proposed NNPEHGP has been successful in avoiding to interpolate the scattered Bayesian residuals on the predicted noise level. It can be seen that the second GP could capture the noise level trend when the Bayesian residuals are smoothed prior to its modelling. These better predictions on the noise level trend yield to a fine predictions on the mean and variance of the true functions, as presented in Figs. Figs. II.3b and II.3d. Both configuration for NNPEHGP appeared to give similarly fine predictions in this difficult sampling plan for IMLHGP. Then, it can be noticed that the proposed modifications in NNPEHGP model leads to a more robust HGP in modeling heteroscedastic problems with a low training sample size. The computational time of NNPEHGP is also still similar to IMLHGP since the distance-based regression procedure is also very cheap and computationally simple. The rest of the procedures of the NNPEHGP is similar to that of IMLHGP as explained in the following section.



(a) Bayesian Residuals Smoothing in NNPEHGP-0.4n.

(b) NNPEHGP-0.4n Prediction.

(c) Bayesian Residuals Smoothing in NNPEHGP-0.2b.

(d) NNPEHGP-0.2b Prediction.

Figure II.3: Examples of NNPEHGP predictions in predicting a one-dimensional heteroscedastic function with a low training sample size. The interval shows 95% error variation of the true function and the GP.

II.3.2 Constructing NNPEHGP

1. Train a standard homoscedastic GP GP_1 on the training samples $D = \{(\boldsymbol{x}_i, y_i)|i = 1, \ldots, n\}$ and estimate the posterior distribution over training outputs as
$$y_i|\boldsymbol{x}_i, \boldsymbol{\theta}_y, D \sim \mathcal{N}(\mu_{y_i}, \sigma^2_{y_i}) \tag{II.43}$$

2. Calculate the Bayesian residuals $r_i = y_i - \mu_{y_i}$ and build a new training dataset D' for the second GP, $D' = \{(\boldsymbol{x}_i, z_i)|i = 1, \ldots, n\}$ with $z_i = |r_i|^v$.

3. Transform $\boldsymbol{z}$ with distance-based regression by averaging its values to the

values in the neighboring inputs using Eq. II.41 or Eq. II.41 as $\hat{z}_i$ and then build a new training dataset $D' = \{\boldsymbol{x}_i, \hat{z}_i\}$

4. Train a second standard homoscedastic GP GP_2 on the new dataset D' and estimate the posterior mean over the v-th absolute moment of Bayesian residuals as

$$\hat{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}_{\hat{z}}, D' \sim \mathcal{N}(\mu_{\hat{z}_i}, \sigma^2_{\hat{z}_i}) \tag{II.44}$$

5. Update the most likely noise levels as

$$\tilde{g}_i^v = \max(0, \mu_{\hat{z}_i} s(v)) \tag{II.45}$$

with $s(v) \approx \sqrt{\pi}/[2^{v/2}\Gamma((v+1)/2)]$

6. Make prediction on future observations with the estimated noise levels as

$$y_*|\boldsymbol{x}_*, \boldsymbol{\theta}_y, \tilde{\boldsymbol{g}}, \tilde{g}_*, D \sim \mathcal{N}(\mu_{y_*}, \sigma^2_{y_*}) \tag{II.46}$$

Chapter III

Research Methodology

This chapter is began by explaining treatment to the computer experiments along with the error metrics used to assess the performance of the models. Continued by explaining the test cases used in this study, consisted of two mathematical functions, two real-world experimental datasets, two real-world analytical problems, and a stochastic simulator.

III.1 Problem Setup

III.1.1 Predictive Performance Assessment

To assess the performance of the GP models, appropriate metrics are required to be chosen carefully in representing the performances of the models. Here, it is desired to know how the models predict the mean, noise level and also the total distribution of the true function across the design spaces. Following (Kersting et al., 2007; Lázaro-Gredilla and Titsias, 2011; Zhang and Ni, 2020), the standardized mean squared error (SMSE) is used to measure the error between the mean of the the true function and the one that predicted by GP models. It can be calculated as:

$$\mathrm{SMSE}(f) = \frac{1}{N}\sum_{i=1}^{N} \frac{(\mu_{f_{*,i}} - f_{*,i})^2}{\mathrm{var}(\boldsymbol{f}_*)} \tag{III.1}$$

where $\mu_{f_{*,i}}$ is the GPs posterior mean and $f_{*,i}$ is the true function value at test input $\boldsymbol{x}_{*,i}$, $\boldsymbol{f}_*$ is the variances of the true function values at all test points $\boldsymbol{X}_* = (\boldsymbol{x}_{*,i}, \ldots, \boldsymbol{x}_{*,N})$, and N is the size of test datasets.

Similarly, to quantify the performance of the models in predicting the noise function, the SMSE with respect to the noise standard deviation can be used, expressed as,

$$\mathrm{SMSE}(g) = \frac{1}{N}\sum_{i=1}^{N}\frac{(\mu_{g_{*,i}} - g_{*,i})^2}{\mathrm{var}(\boldsymbol{g}_*)} \tag{III.2}$$

where $\mu_{g_{*,i}}$ is the GPs posterior mean of the noise standard deviation and $g_{*,i}$ is the true noise standard deviation at test input $\boldsymbol{x}_{*,i}$, and $\boldsymbol{g}_*$ is the variances of the true noise standard deviation values at all test points $\boldsymbol{X}_* = (\boldsymbol{x}_{*,i}, \ldots, \boldsymbol{x}_{*,N})$.

To quantitalively measure the performance of the GP models in predicting the full distribution, ε is used as the error measure (Zhu and Sudret, 2020) defined as

$$\epsilon = \mathrm{E}[d_{\mathrm{WS}}(Y(\boldsymbol{X}_*), \hat{Y}(\boldsymbol{X}_*))] \tag{III.3}$$

where $Y(\boldsymbol{X}_*)$ is the true response, $\hat{Y}(\boldsymbol{X}_*)$ is the response from the GP models, and d_{WS} denotes the disparity between the probability distributions of $Y(\boldsymbol{X}_*)$ and $\hat{Y}(\boldsymbol{X}_*)$. Here, d_{WS} is the Wasserstein distance of order two (Villani, 2008) defined by

$$d_{\mathrm{WS}}(Y_1, Y_2) \overset{\mathrm{def}}{=} ||Q_1 - Q_2||_2 = \sqrt{\int_0^1 (Q_1(u) - Q_2(u))^2 \, \mathrm{d}u} \tag{III.4}$$

where Q_1 and Q_2 are the quantile functions of Y_1 and Y_2, respectively.

However, the true function, along with its noise standard deviation, in all cases could not be known or empirically estimated. For example, in the motorcycle and LIDAR experimental data used in Sect. III.3, only limited experimental observations are available. Hence, it is not possible to calculate the SMSE and ε metrics. To assess the performance of the GP models on these real-world experimental datasets, the average negative log probability density (NLPD) of test outputs is used to measure the accuracy of the GP models in predicting the data distributions. NLPD can be computed as,

$$\begin{aligned}
\mathrm{NLPD}(y) &= -\frac{1}{N}\sum_{i=1}^{N}\log p(y_{*,i}|\boldsymbol{x}_{*,i}, D) && \text{(III.5)}\\
&= \frac{1}{2N}\sum_{i=1}^{N}\log(2\pi\sigma^2_{y_{*,i}})\frac{1}{N}\sum_{i=1}^{N}\frac{(y_{*,i} - \mu_{y_{*,i}})^2}{2\sigma^2_{y_{*,i}}} && \text{(III.6)}
\end{aligned}$$

where $\mu_{y_{*,i}}$ and $\sigma^2_{y_{*,i}}$ is the GPs posterior mean and variance at each test point

$\boldsymbol{x}_{*,i}$. $y_{*,i}$ is the sample's value at each validation input $\boldsymbol{x}_{*,i}$ and n_{val} is the size of validation data sets. Lower losses in all of the error metrics used here indicate better predictive performance of the models.

III.1.2 Problem Treatment

In the present work, the problems shall be treated as follows:

- The number of training samples at each test case is varied into three different sizes variations – small, moderate, and large. The variation of each training sample sizes is determined by accommodating the problems' complexity. The traning sample size is mainly varied as: $20m$, $40m$, and $80m$, where m is the number of dimension in the problem. In the real-world experimental data, the training sample is varied as: 50%, 70%, and 90% of the total samples.
- For the real-world analytical problems and stochastic simulator, the true noise level and the true samples' distribution is empirically estimated by conducting a large number of replications at each test input. The number of replications is chosen by considering the computational cost of each problem.
- Each sample variation has 20 different sample positions with each representing the performance metrics of each model. If the location of samples can be generated, i.e., other than real world experimental data sets, the training samples is generated by using a Latin Hypercube Sampling (LHS) plan.
- The surrogate models used in the computational experiment is consisted of the proposed Nearest Neighbor Point Estimates HGP (NNPEHGP) model with four parameter variations, Improved Most Likely Heteroscedastic GP (IMLHGP), and the standard homoscedastic GP.
- The performance metrics of the surrogate models will be mainly compared by visualizing them as a boxplot for 20 error values.

III.2 Mathematical Function Test Cases

On the performance of the surrogate models, the computational experiment starts by using the mathematical function as the test case problems. This strategy is chosen since the equation can be easily defined with certain characteristic of the function. The mathematical function test case problems used in this works consists of a one-dimensional mathematical function and a two-dimensional mathematical function. Both of the functions have an input-varying noise level and also have nonlinear characteristics in the mean response to represent usual engineering responses.

III.2.1 One-Dimensional Mathematical Function

Following Bae, Clark, and Forster, 2019, the one dimensional mathematical function used in this study is expressed as,

$$y = x + x\sin^2(2\pi(x-3)) + 2\exp(x-2) + 0.2x\zeta, \tag{III.7}$$

where x is the input with the range $[0 \sim 1]$ and ζ is the standard normal random variable $\mathcal{N}(0,\,1)$. This 1-D function has a nonlinear mean function and noise strength that increases linearly as a function of x, visualized in Fig. III.1.

This one-dimensional mathematical function is chosen as the first problem to assess the performance of the models since its low dimensionality makes the function relatively easy to model. The predictive performance of the models in learning this function can also be easily visualized to give insights about the characteristics on each surrogate model's prediction. Hence, the study is not only to see which model performs better but also the corresponding respective phenomena. The number of training samples in this problem is chosen as 20, 40, and 80 samples.

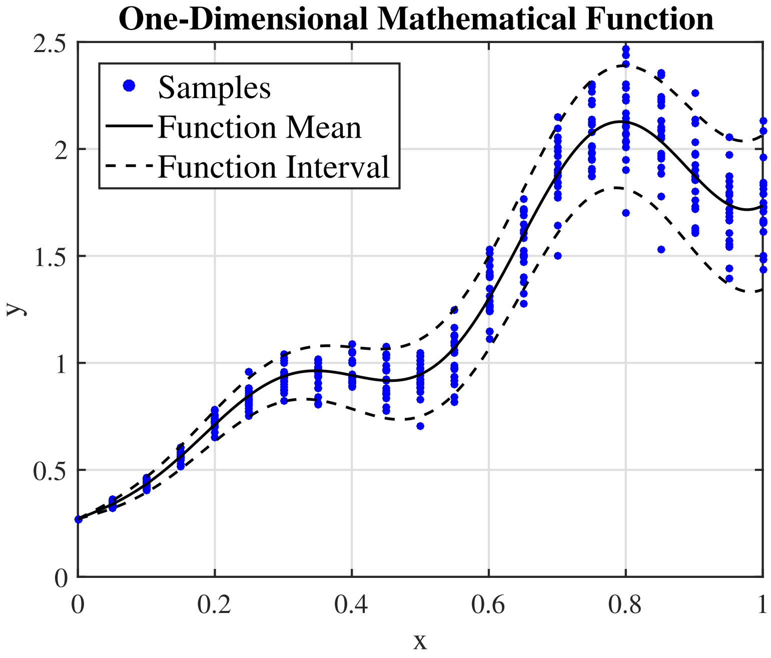


Figure III.1: One-dimensional mathematical function used in this study. The interval shows 95% error variation of the function, that is defined with the noise level at each input. Several samples is also displayed (as blue dots) to show the heteroscedastic characteristic of the function.

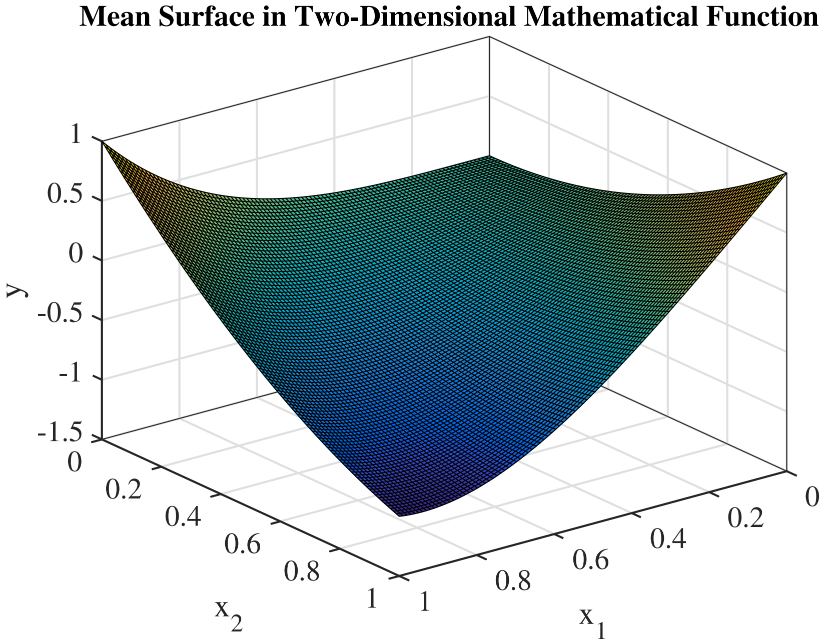


Figure III.2: Mean surface in the two-dimensional mathematical function used in this study. Notice the nonlinear characteristics in the surface.

III.2.2 Two-Dimensional Mathematical Function

To see the performance of the surrogate models in learning higher dimensional heteroscedastic problem, a two-dimensional mathematical function is chosen as the second problem in this section. Although the dimension remains relatively low, it is important to see the performance of the models in modelling a two-dimensional problem since IMLHGP–as the benchmark model – have not shown its performances in problems that has more than one dimension.

The two-dimensional mathematical function used here is also following Bae, Clark, and Forster (2019), expressed as,

$$y = x_1^{3.5} + x_2^2 - 3x_1x_2 + 0.1x_2\zeta \tag{III.8}$$

where x_1 and x_2 are the inputs within $[0 \sim 1]$ while ζ is the standard normal random variable $\mathcal{N}(0,\, 1)$. This function also has a nonlinear characteristics in the mean function trend in the design spaces (as shown in Fig. III.2) and a linearly increasing noise strength as a function of the second variable of the inputs x_2 (as shown in Fig. III.3). In training the model to learn this function, the number of training samples are chosen as 40, 80, and 160 samples.

III.3 Real-World Experimental Data

In this section, the performance of the surrogate models, in learning heteroscedastic samples observed from real-world experiments, is presented. The dataset chosen in this study consists of two benchmark experimental data that is usually used in heteroscedastic GP (HGP) study. The performance of the surrogate models, in learning real world experimental data, would present interesting observations since it represents the future practical application in learning samples obtained from physical experiments.

The first example of the experimental data is the LIDAR (LIght Detection And Ranging) dataset (Sigrist, Bernegger, and Meyer, 1989) that consists 221 observations from a light detection and ranging experiment. The logarithm of the ratio

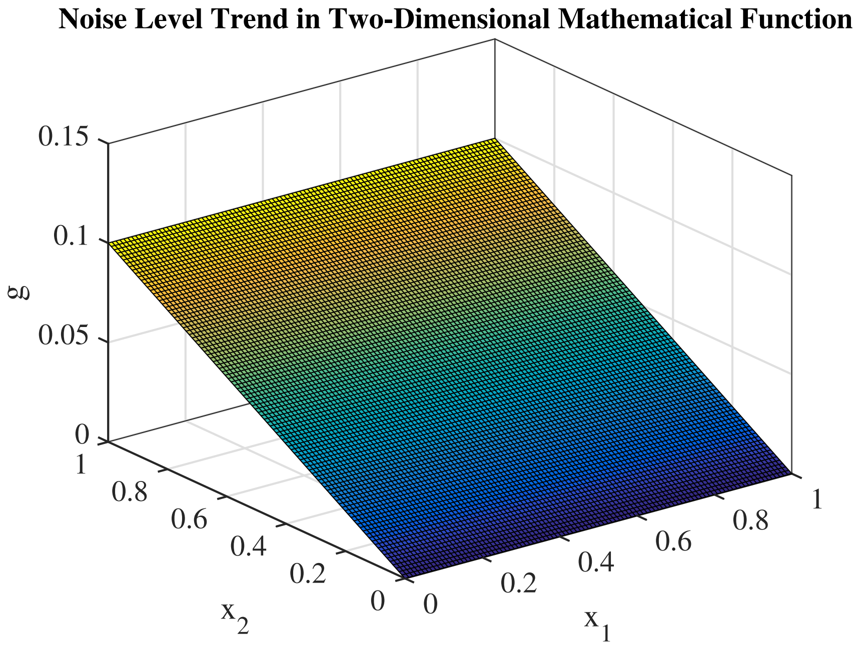


Figure III.3: Noise level trend in the two-dimensional mathematical function used in this study. The trend is shown in its noise standard deviation value, that is linearly increasing with the second input x_2.

of received light from two laser sources are observed for several distances traveled before the light is reflected back to its source. This data is visualized in Fig. III.4. In training the model, the proportions training samples is chosen as: 50%, 70%, and 90% of total observations.

The second example of heteroskedastic experimental data is the motorcycle accident dataset (Silverman, 1985), a classic example that have been used in many works (Binois, Gramacy, and Ludkovski, 2018; Lázaro-Gredilla and Titsias, 2011; Zhang and Ni, 2020). This data consists of 133 observations that explain human head's accelerations with respect to time in simulated motorcycle crashes. The objective of this experiment is to determine the efficacy of crash helmets. The experiments are conducted at 94 unique time inputs that has more than one realizations at several locations. This experimental data is visualized in Fig. III.5. It can be observed that this experimental data is rather complex than the previous LIDAR experimental data with a nonlinear trend and an extremely varying noise strength.

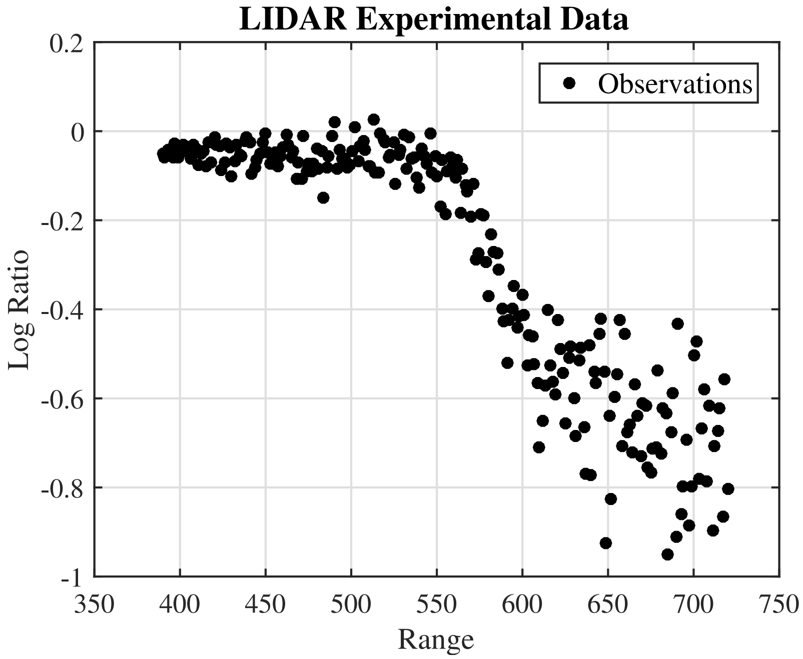


Figure III.4: LIDAR experimental data. The heteroscedasticity in the dataset can be observed as the variability of the samples increased with the input.

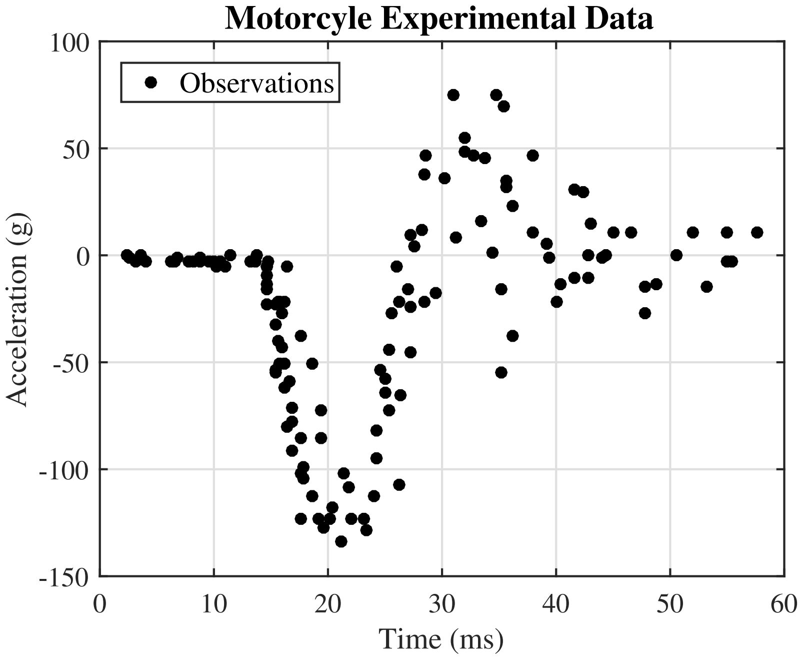


Figure III.5: Motorcycle experimental data. Notice the complex varying noise level as a sudden high variability in the samples can be observed in the middle of the input spaces.

Previous works modelling this data mostly uses 90% of the total data as training, including the replicated samples; and the rest to be test samples. In this thesis, the proportion of the training samples is chosen to be 50%, 70%, and 90% of the unique time input to show the performance in varying number of training samples. This work also avoids to use replications in training the model to remain consistent in the non-replication modelling framework.

III.4 Real-World Analytical Problems

Consequently, in the present section, heteroscedastic data observed from higher dimensional real-world analytical problems are studied. The problems in this study consist of a three-dimensional operational envelope analysis of a generic transportation aircraft and a four-dimensional temperature modeling of a solid immersed in fluid. The heteroscedasticity in the samples is achieved by considering several parameters in each equation as a random variable.

III.4.1 Three-Dimensional Operational Envelope Analysis of A Generic Transportation Aircraft Under Uncertainty

The first example of high-dimensional aerospace problems is the operational envelope study of a generic transportation aircraft. The Breguet range equation (Raymer, 2006) is often used in a conceptual design study of airplane size and configurations in the preliminary performance study. The equation is expressed as,

$$W_{fuel} = (W_e + W_p)\exp\left(\frac{R \cdot SFC}{V \cdot LOD} - 1\right) \tag{III.9}$$

Here, W_{fuel}, W_e, and W_p are the fuel, payload, and empty vehicle weights (in pounds), respectively. R is the range of an aircraft (in nautical miles), SFC is the aircraft's specific fuel consumption rate (one per hour), V is the aircraft velocity (in miles per hour), and LOD is the lift-to-drag ratio (pound/pound).

Table III.1: Specifications of generic transportation aircraft in the operational envelope problem.

Specifications	Generic Value
Fuel weight W_{fuel}, lb	208,480
Empty vehicle weight W_e, lb	282,500
Maximum payload W_p_max, lb	170,900
Maximum range R_max, miles	2,785
Maximum velocity V_max, miles/h	515
Nominal specific fuel consumption SFC, gal/h	0.66
Nominal lift-to-drag ratio LOD, lb/lb	16

The Breguet range equation essentially estimates how much fuel needed by an aircraft to complete a mission described by its mission and configuration parameters: weight of the aircraft, weight of the payload to deliver, distance to travel along with its velocity, engine performance, and aerodynamic efficiency. This is estimated by assuming that the aircraft is operating at a constant airspeed and fixed aerodynamic efficiency (LOD) in retained specific altitude and weight profile. It is then appropriate to use this equation in a steady-state cruise profile of flight, mainly for a commercial transportation aircraft.

In this example, the generic transportation aircraft, as specified in Table III.1 is studied while taking note that there may exist a randomly perturbation. The random perturbation is defined by treating several parameters, that is SFC and LOD as random variables, following the works in Bae, Clark, and Forster (2019). This perturbation represents uncertainties in its operational condition that might happen due to inconsistent environmental and operational conditions, such as a change due to gust, different weather conditions, and fluctuating cruising altitudes. The random perturbations are modeled as normal distributions around its mean values with about a 3% coefficient of variance. The aircraft will be tested for wide ranges of missions that are determined by payload weights W_p, distances R, and times to deliver V as the inputs. The configuration for the parameters studied in this thesis is shown in Table III.2.

The Breguet range equation here is treated as a virtual black-box simulation to asses the performance of this generic transportation aircraft, by observing the estimated values of fuel weights W_{fuel} needed by the aictaft at each design con-

Table III.2: Operational and random parameter ranges for the three-dimensional operational envelope problem.

Variables	Ranges/distributions
Payload	$[0.1 \sim 0.9] \times W_p$_max
Range	$[0.7 \sim 1.0] \times R$_max
Velocity	$[0.9 \sim 1.1] \times V$_max
SFC	$\mathcal{N}(0.66, 0.02)$
LOD	$\mathcal{N}(16.0, 0.53)$

figuration. Due to the randomness in SFC and LOD, the observed samples at a design configuration is also uncertain with unknown true mean and variances. To assess the performance of the GP models, test samples is obtained by conducting experiments at 8000 number input locations, each with 10^5 realizations. The true variability of the samples, defined by its noise level, then can be well estimated while the true mean function is obtained by fixing the random parameters, i.e., SFC and LOD, at its mean values. In training the surrogate models, the number of training samples is varied as 60, 120, and 240.

III.4.2 Four-Dimensional Temperature Modeling of A Solid Sphere Immersed in Fluid Under Uncertainty

The second real world analytical problem is a temperature modeling of a solid sphere immersed in fluids, following the works in Tan (2016). This problem is mainly a heat transfer problem that is previously described by Cengel and Heat (2003), where the solid sphere is immersed in a fluid with higher initial temperature. Heat is then transferred between the sphere and the fluid via convection and inside the sphere itself via conduction process. The temperature Y of the solid sphere is described as the solution of a partial differential equation, given explicitly by

$$
\begin{aligned}
Y = u_3 + u_4 \sum_{i=1}^{\infty} \frac{4\left(\sin \eta_i - \eta_i \cos \eta_i\right)}{2\eta_i - \sin\left(2\eta_i\right)} \exp\left(-\eta_i^2 \pi_2\right) \frac{\sin\left(\eta_i u_1\right)}{\eta_i u_1}, \\
1 - \eta_i \cot \eta_i = \pi_1, \eta_i \in ((1-i)\pi, i\pi), \\
\pi_1 = \frac{u_5 u_6}{u_7}, \pi_2 = \frac{u_6 u_2}{(u_8 u_9 u_7^2)}
\end{aligned} \tag{III.10}
$$

where $u_1, \ldots, u_9$ is the nine inputs listed in Tables III.3 and III.4, π_1 is the Biot number, and π_2 is the Fourier number. The η_i are the solutions of the equation $1 - \eta_i \cot \eta_i = \pi_1$, where $\eta_i \in ((1-i)\pi, i\pi)$. Following Tan, 2016, the infinite series in Eq. III.10 are approximated with the sum of the first four terms. This termperature model is known to be useful for applications in manufacturing process design (Jaluria, 2002).

In this thesis, the temperature modelling of the solid sphere is studied by considering four input parameters. These input parameters are defined as the distance from the center of the sphere u_1, the temperature of the fluid u_2, the initial temperature of the sphere u_3, and the convective heat transfer coefficient u_4. From a physical viewpoint, the temperature of the sphere T is increasing in $u_1, \ldots, u_4$ whenever the fluid temperature is higher than the sphere's initial temperature.

For this problem, in representing the operational uncertainties, the $u_6, \ldots, u_9$ are treated as unknown random variables defined with a normal distribution. The parameter of time u_5 in the equation is defined as a fixed value in this study. This fixed value, i.e. 800 s, of the time parameter u_5 is chosen after observing the distributions of the solid's temperature to be approximately Gaussian at this specific time; hence can be modeled by GP surrogate models used in this study. Using this configuration, the observations at each input location will also be uncertain with unknown true mean and variances. In this study, the problem here is treated in its standardized form of the input parameters, $z_1, \ldots, z_4 \in [0,1]^4$ where $z_i = (u_i - \max(u_i))/(\max(u_i) - \min(u_i))$.

Due to a more expensive problem than the previous operational envelope study problem (Sect. III.4.1), fewer test samples and realizations are used to estimate the true mean and variances. The performance of the GP models will be later assessed by comparing the predictions to the reference values in 1000 test samples with 10^4 realizations at each test input location. The number of training samples in this problem are varied as 80, 160, and 320 samples.

Variable	Description	Units	Minimum	Maximum
u_1	Distance from the center of sphere	-	0	1
u_2	Temperature of fluid	K	250	270
u_3	Initial sphere temperature	K	-100	-30
u_4	Convective heat transfer coefficient	$kgs^{-3}K^{-1}$	180	210

Table III.3: Ranges of input parameters for the four-dimensional temperature modeling problem.

Variable	Description	Units	Values / Distributions
u_5	Time	s	800
u_6	Thermal conductivity	$kgs^{-3}K^{-1}$	$\mathcal{N}(65,\ 10.3^2)$
u_7	Radius of sphere	m	$\mathcal{N}(0.1,\ 0.02^2)$
u_8	Specific heat	$m^2s^{-2}K^{-1}$	$\mathcal{N}(400,\ 33.3^2)$
u_9	Density	kgm^{-3}	$\mathcal{N}(8000,\ 333.3^2)$

Table III.4: Fixed and random parameter configurations for the four-dimensional temperature modeling problem.

III.5 Stochastic Simulator Test Case

On the performance of the surrogate models in learning heteroscedastic data from a stochastic simulator, a Feynman-Kac simulator (Herbei and Berliner, 2014) is used as the stochastic simulator in this study. This Feynman-Kac simulator is described as a simulator to model the concentration of a tracer within a defined spatial domain. In this study, following the examples in Herbei and Berliner (ibid.), this simulator is used to model oxygen concentration in a thin water layer deep in the ocean, based on the study by McKeague et al. (2005).

This simulator works by approximating the solution of an advection-diffusion process with Monte Carlo (MC) procedure, providing stochastic samples in the input spaces. Thus, unlike the deterministic simulator, this simulator might generate samples with different values at a set of input locations. Samples generated from

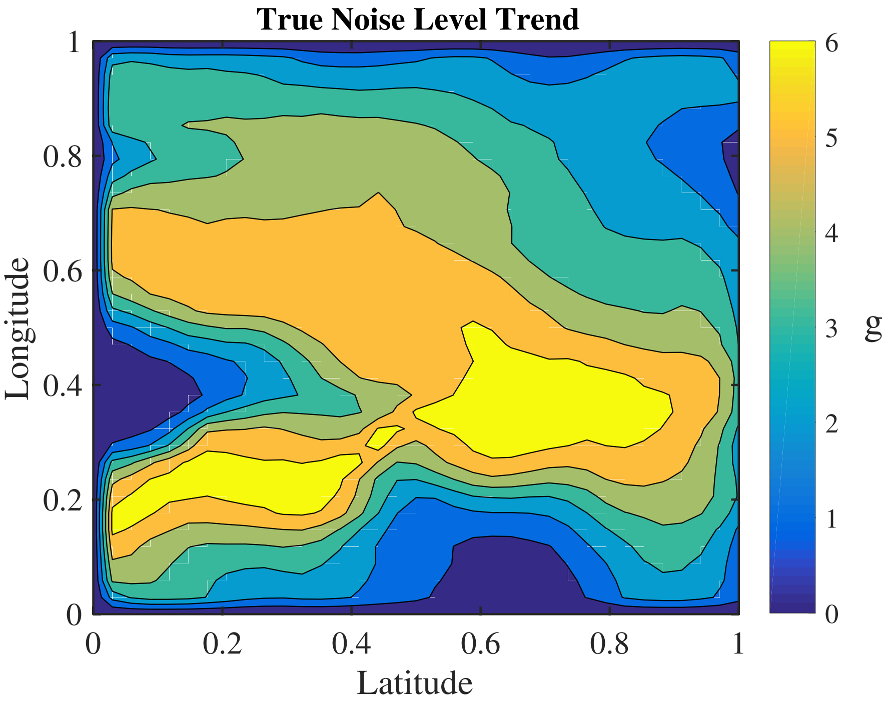


Figure III.6: Noise level trend in the stochastic simulator used in this study. The trend is shown in its noise standard deviation value, that is varying in the input spaces.

this simulator is observed to be highly heteroscedastic with its noise level is visualized in Fig. III.6. The majority of the samples obtained from this simulator is also observed to be non-Gaussian distributed such that the GP models used in this study may be difficult in learning the distribution correctly. Since the available stochastic simulators are quite few, this study, still, attempts to use this example of a stochastic simulator as it represents one application of the models. It also provides an insight into the performances of the model in learning a non-Gaussian distributed samples.

In this experiment, the inputs have two parameters that defines the spatial coordinates, i.e., longitude and latitude. The code for this stochastic simulator is provided by Gramacy (2020), available in R programming language. In training the models to learn this stochastic simulator, the number of samples are varied as: 40, 80, and 160. The performance of the GP models will be assessed by comparing the predictions to the reference values in 1225 test inputs, with 10^4 realizations at each test input location.

Chapter IV

Surrogate Models Performance

In this chapter, the performance of each surrogate model is analyzed and compared with one another. The framework for assement, of present, will be based on the resulting boxplots. With the aid of the boxplot, specifically, the distribution of each surrogate model's performance is used to explain the characteristics, such as robustness, of the latter. In the proceeding boxplot figures, the standard homoscedastic GP is denoted as HomGP and the improved most likely HGP (IMLHGP) is labelled as IML. Furthermore, the model proposed in the present thesis, Nearest Neighbor Point Estimates HGP, is labelled as NNPE, followed by the noise smoothing configurations. Namely, $k = 0.2n$ and $k = 0.4n$ if the configuration is based on the number of the nearest neighbors and $d = 0.2b$ and $d = 0.4b$ if it is based on the maximum distance.

The error metrics discussed in the present section is the SMSE(g) and ε for the mathematical functions, real-world analytical problems, and the case of stochastic simulator. Moreover, the performances of the surrogate models over the real-world experimental data is compared by investigating the NLPD metrics. The performances in predicting the true mean function, i.e., SMSE(f), is not discussed in this chapter since the performance of all models viewed from this standpoint is found to have no significant difference, presented in Appendix. The similar performances in predicting the mean function is caused by the procedure of all models to learn the true mean function is relatively similar, by using a standard homsocedastic GP. Furthermore, the influence of the difering noise matrices across the models is also shown to be insignificant in predicting the mean function.

Lastly, it is also desired to investigate the phenomena behind the predictive performances to give insights about the characteristics of the surrogate models. With this regard, the visualization of the predictions of several problems is also presented. The analysis based on this visualizations will be focused on our proposed NNPEHGP model compared with the corresponding benchmark IMLHGP model.

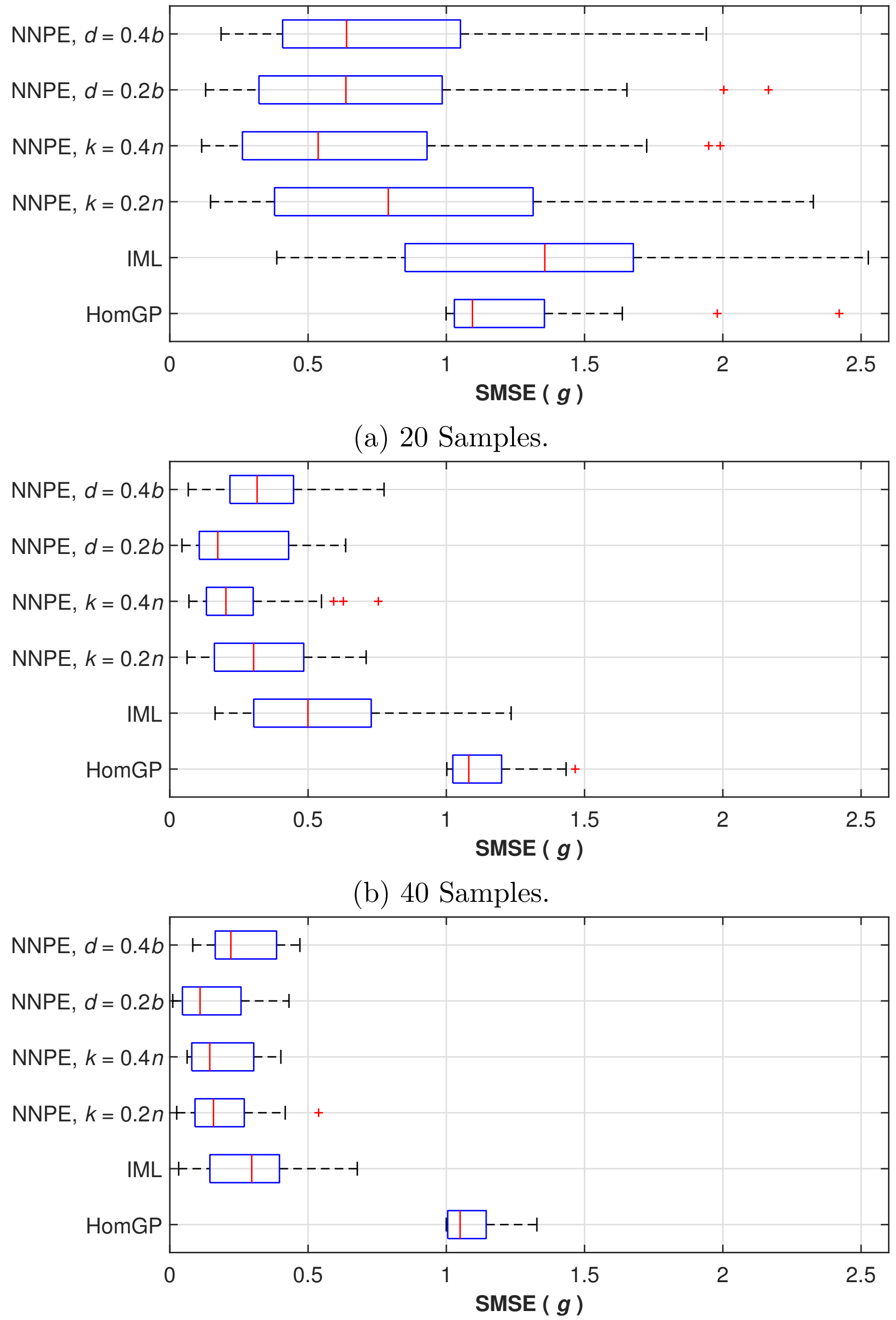
IV.1 Analytic Test Case Results

IV.1.1 One-Dimensional Mathematical Function

The performances of the model in predicting the noise standard deviation (i.e., SMSE (g)) in the one-dimensional mathematical function is presented in Fig. IV.1. With a small training sample size $(n = 20)$, as observed in Fig. IV.1a, the performance of IMLHGP in predicting the noise level yields worse results, compared with a standard homoscedastic GP. On the other hand, the proposed NNPEHGP shows an improved performance in predicting the current noise level. Furthermore, the latter model, utilizing the number of nearest neighbors $(k = 0.4n)$ demonstrates an improved result compared with the other configurations. What is to be obtained from these results, is the fact that each one of the NNPEHGP configurations shows a better prediction compared to the IMLHGP and standard homoscedastic GP, with several bad results that seem to appear for a few unlucky samples.

Consequently by observing the predictive performance on the noise standard deviation with a moderate traning sample size $(n = 40)$, as visualized in Fig. IV.1b. The performance of the IMLHGP improves and further beats the standard homoscedastic GP. However, it is shown that the worst result of the latter appears to be worse than the overall worst result of the standard homoscedastic GP, which essentially tells us that the performance of the model is not quite robust. Furthermore, it is noted that each configuration of NNPEHGP shows more robust predictions compared to IMLHGP where not a single result indicates any values worse than the standard homoscedastic GP. The best performer in the present case is the NNPEHGP model with $k = 0.4n$.

For a large training sample size $(n = 80)$, the performances of IMLHGP and NNPEHGP can be observed to be similarly good in predicting the noise standard



(a) 20 Samples.

(b) 40 Samples.

(c) 80 Samples.

Figure IV.1: SMSE (g) results for one dimensional mathematical function.

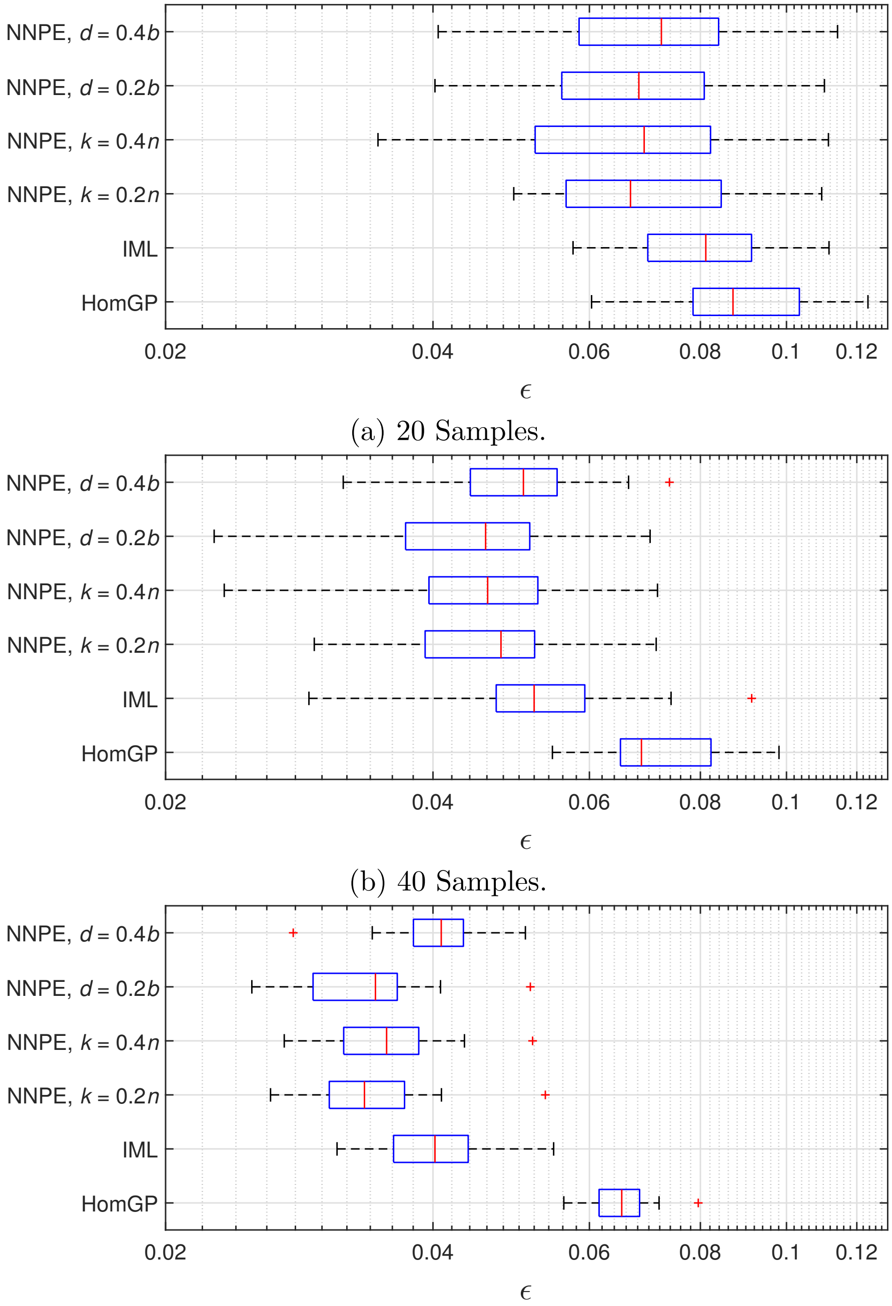
deviation, as shown in Fig. IV.1c. From this result, it can be inferred that IML-HGP requires a greater number of samples to have good prediction on the varying noise level. Here, it can also be noticed that NNPEHGP with $k = 0.2n$, $k = 0.4n$, and $d = 0.2b$ is shown to have almost identical performances, wherein they show to slightly improve in accuracy and robustness, compared to the performances of IMLHGP.

The performance results on predicting the full distributions of the function, in terms of ε, is shown in Fig. IV.2. Here, it can be observed at each training sample size, IMLHGP yields a significantly better prediction compared to the standard homoscedastic GP from the perspective of ε. Observing the error results in the low training sample size ($n = 20$), as visualized in Fig. IV.2a, each configuration of NNPEHGP showed to have a lower median than the IMLHGP. It can also be observed, with a low training sample size, NNPEHGP with $k = 0.4n$ achieved a much better result in predicting the heteroscedastic function distributions.

With a moderate training sample size, as visualized in Fig. IV.2b, NNPEHGP is observed to have more accurate prediction in the function distributions as opposed to the IMLHGP. It can also be seen that IMLHGP has a poor outlier performance. The results for as high training sample size, as shown in Fig. IV.2c, also show interesting insights into the performances of the GP models. Here, IMLHGP as expected, clearly shows a much better performance than the standard homoscedastic GP. Yet, it can be observed that the performances of NNPEHGP, particularly ones with $k = 0.2n$ and $d = 0.2b$, yielded significantly better performances than IMLHGP. Therefore, even in high training sample size, the proposed NNPEHGP still might be preferred to model heteroscedastic problems.

To further investigate the difference in performance between IMLHGP and NNPEHGP, the best and the worst prediction from the two models are visualized in Figs. IV.3 and IV.4. The definition of the best and worst results chosen here is according to the Wasserstein distance. The results visualized here is the ones with moderate training sample size to represent the performance when only quiet few samples are available. The configuration for NNPEHGP here is chosen to be NNPEHGP with $k = 0.4n$ as it shows the best performance for the NNPEHGP.

As seen in Fig. IV.3, the two models perform quite similarly at each of their best conditions. However, it can be noticed that NNPEHGP captures the varying noise level within the low variance region better than the IMLHGP. The difference between the two methods is more prominent in their worst conditions, in which,

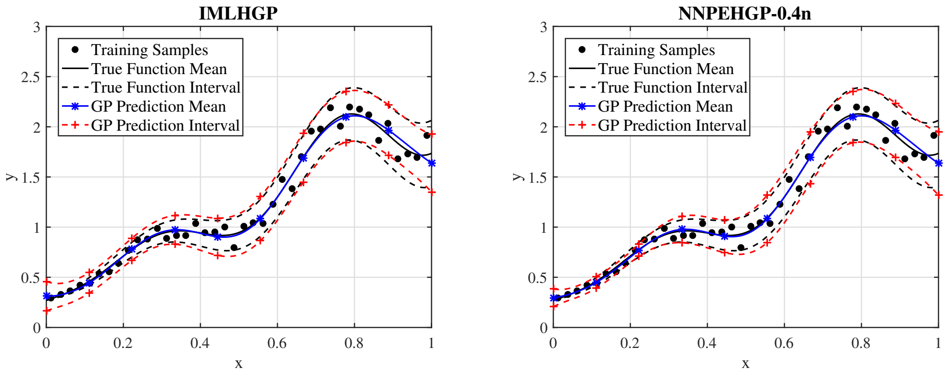


(a) 20 Samples.

(b) 40 Samples.

(c) 80 Samples.

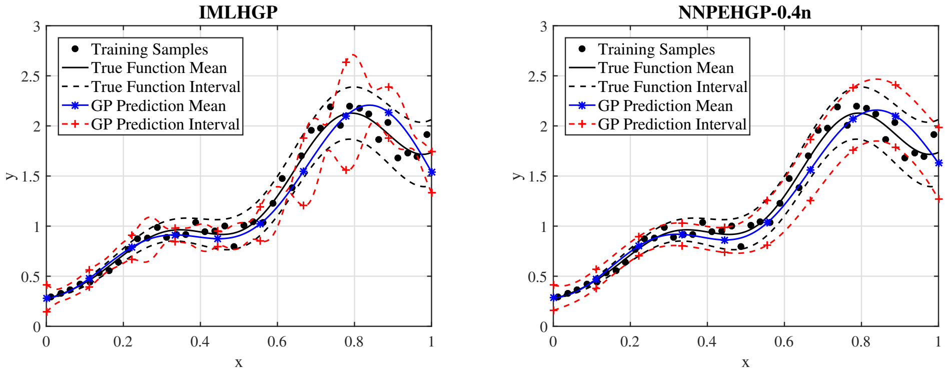
Figure IV.2: Wasserstein distance results for one dimensional mathematical function.



(a) IMLHGP.

(b) NNPEHGP.

Figure IV.3: The best IMLHGP and NNPEHGP ($k = 0.4n$) models in terms of the Wasserstein distance for the one-dimensional mathematical function with $n = 40$. The interval shows 95% error variation of the true function and the GP.



(a) IMLHGP.

(b) NNPEHGP.

Figure IV.4: The worst IMLHGP and NNPEHGP ($k = 0.4n$) models in terms of the Wasserstein distance for the one-dimensional mathematical function with $n = 40$. The interval shows 95% error variation of the true function and the GP.

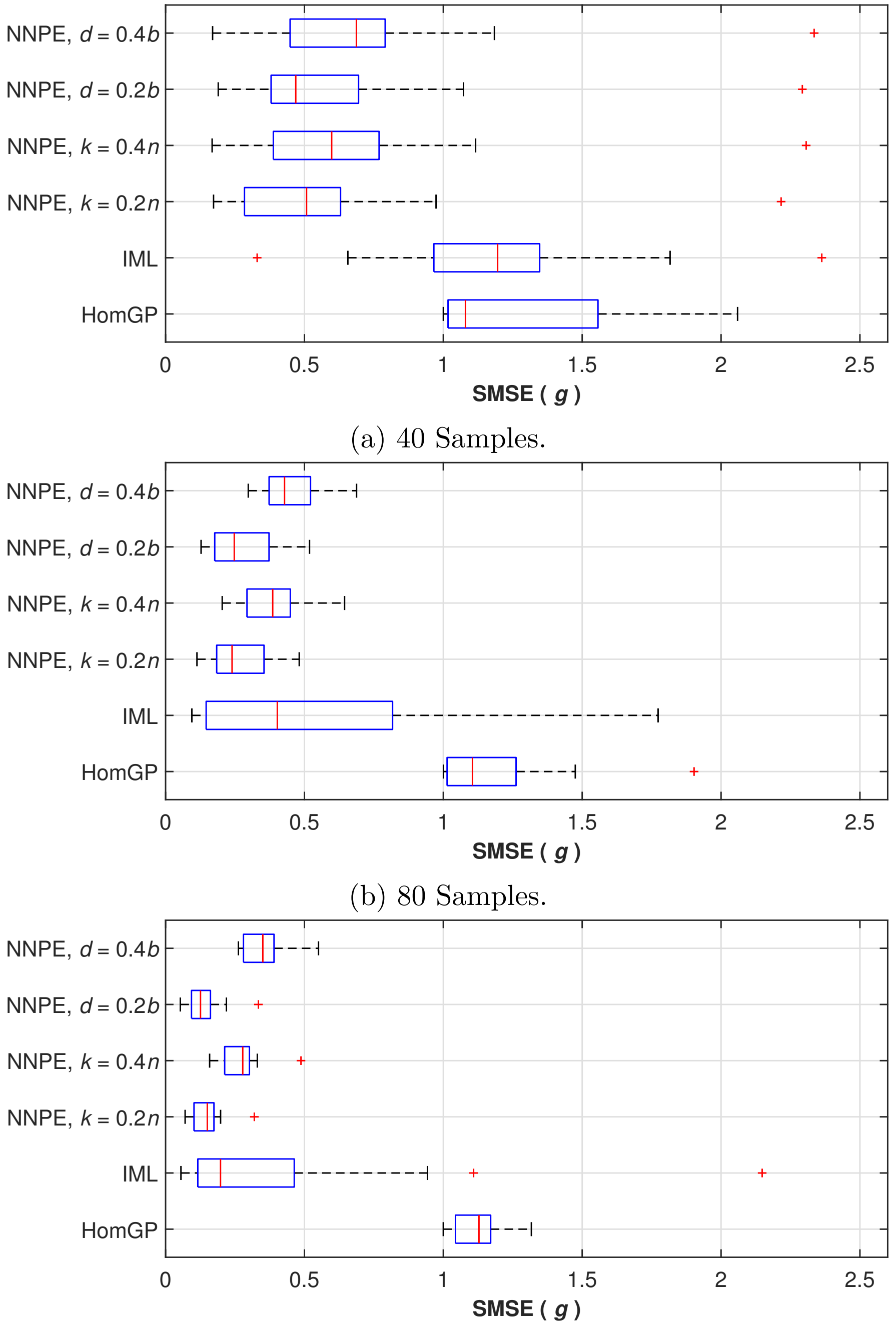
the IMLHGP fails in learning the true noise level as it seems to interpolate the scattered Bayesian residuals in this unlucky sampling condition. While the prediction from NNPEHGP still manages to capture the increasing noise level in the input spaces. This one-dimensional mathematical function results suggest that the performance of IMLHGP is rather unstable when the number of training samples is low. The proposed NNPEHGP model can be observed to avoid interpolates the Bayesian residuals in estimating the noise level; this explains the better and more robust predictive accuracy compared to the benchmark IMLHGP model.

IV.1.2 Two-Dimensional Mathematical Function

The SMSE (g) results for the two-dimensional mathematical function as shown in Fig. IV.5 demonstates that, in Fig. IV.5a, as the result of one-dimensional function, IMLHGP performed poorly in predicting the varying noise level when the training sample size is small ($n = 40$). It is seen that the performances of IMLHGP remains worse than the standard homoscedastic GP. Note a rather small increasing noise level trend in this function, as descrived in Eq. III.8, even IMLHGP fails to predict this in the lowl training sample size. In this regard, it can be observed that NNPEHGP shows significantly better results than IMLHGP and standard homoscedastic GP; where NNPEHGP with $k = 0.2n$ and $d = 0.2b$ shows the best results. However, with a small training sample size, both IMLHGP and NNPEHGP shows a similar poor outlier in a situation of unlucky sampling.

For the moderate and high training sample size, i.e., $n = 80$ $n = 160$ respectively, as visualized in Figs. IV.5b and IV.5c, a similar trend in the performances of the models can be observed. Here, IMLHGP shows, generally, better results than the standard homoscedastic GP, which implies that the IMLHGP is already successful in capturing the varying noise level. However, the performances of IMLHGP can be observed to not be robust, particularly in the moderate training sample size (Fig. IV.5b) where IMLHGP has plenty of poor results that is worse, compared to the overall performances of the standard homoscedastic GP. While IMLHGP improves for a high training sample size, there is no indication of robustness in its performance. Furhtermore, for the moderate and high training sample size, it can be observed that NNPEHGP shows a more robust result than the IMLHGP and a better performance in predicting the noise standard deviation than the other models. The IMLHGP's lack of robustness showed in these results is really interesting to notice. In their original paper (Zhang and Ni, 2020), IMLHGP has not been tested on modeling high dimensional problems. And as can be observed in this two-dimensional mathematical function, even in the high training sample size, the performances of IMLHGP in the predicting the noise level is unstable.

The best and the worst noise level predictions for IMLHGP and NNPEHGP ($k = 0.2n$) for the two-dimensional mathematical function in moderate training sample size ($n = 80$) is shown in Figs. IV.6 and IV.7. The significant difference can



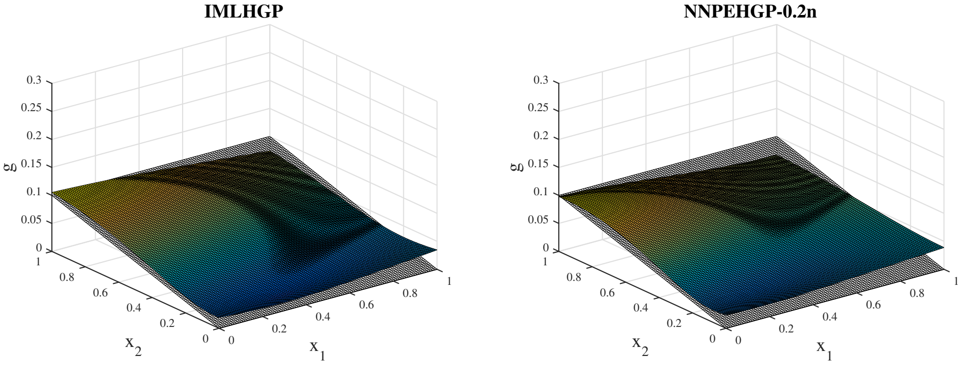
(a) 40 Samples.

(b) 80 Samples.

(c) 160 Samples.

Figure IV.5: SMSE (g) results for two dimensional mathematical function.

be observed particularly in their worst predictions (Fig. IV.7). Here, it can be observed that the worst prediction from IMLHGP is very poor, even when the linearly increasing trend of the noise level is quite small. Contrast to the results



(a) IMLHGP.

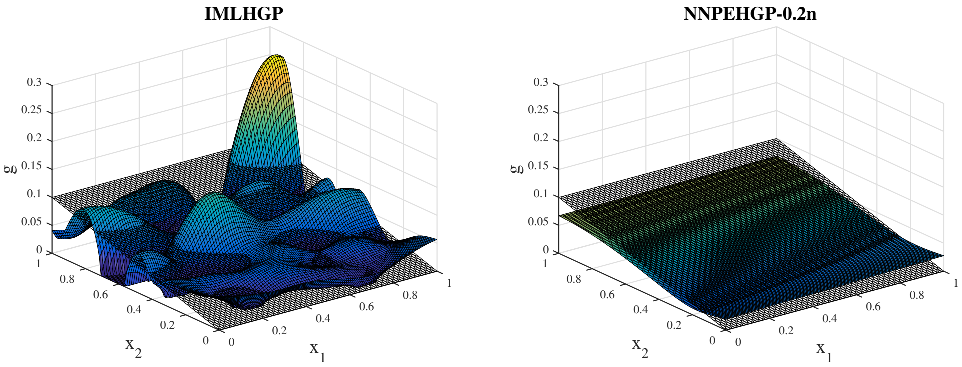
(b) NNPEHGP.

Figure IV.6: The best IMLHGP and NNPEHGP ($k = 0.2n$) models in predicting the noise level for the two-dimensional mathematical function with $n = 80$. The transparent figures are the true noise standard deviation, while the filled figures are the predicted noise standard deviation.

of NNPEHGP, which manages to maintain a good prediction in capturing the varying noise level.

The performances of the model in predicting the function distributions (in terms of ε) are presented in Fig. IV.8. For a low training sample size, as showed in Fig. IV.8a, IMLHGP, when predicting the distributions of the function, demonstrates a smilar performance to that of the standard homoscedastic GP. This is undersirable because this indicates the IMLHGP unable, to capture the varied distributions in the input spaces since the standard homoscedastic GP is designed to learn a constant distributions' variance. Similarly, with the resuls in the error of noise level prediction (Fig. IV.5a), for a low training sample size, it can be observed that NNPEHGP performed better than both IMLHGP and standard homoscedastic GP.

For a moderate training sample size, as shown in Fig. IV.8b, it is observed that IMLHGP may achieve one of the best results in predicting the function's distribution, but still, robustness in the results is not shown. Moreover, it can also be observed that NNPEHGP, particularly ones with $k = 0.2n$ and $d = 0.2b$, achieves a lower median in its performances with more robust results than the IMLHGP. The results for a large training sample size, as shown in Fig. IV.8c, indicates that the IMLHGP and NNPEHGP achieves a similar result in their optimum,



(a) IMLHGP.

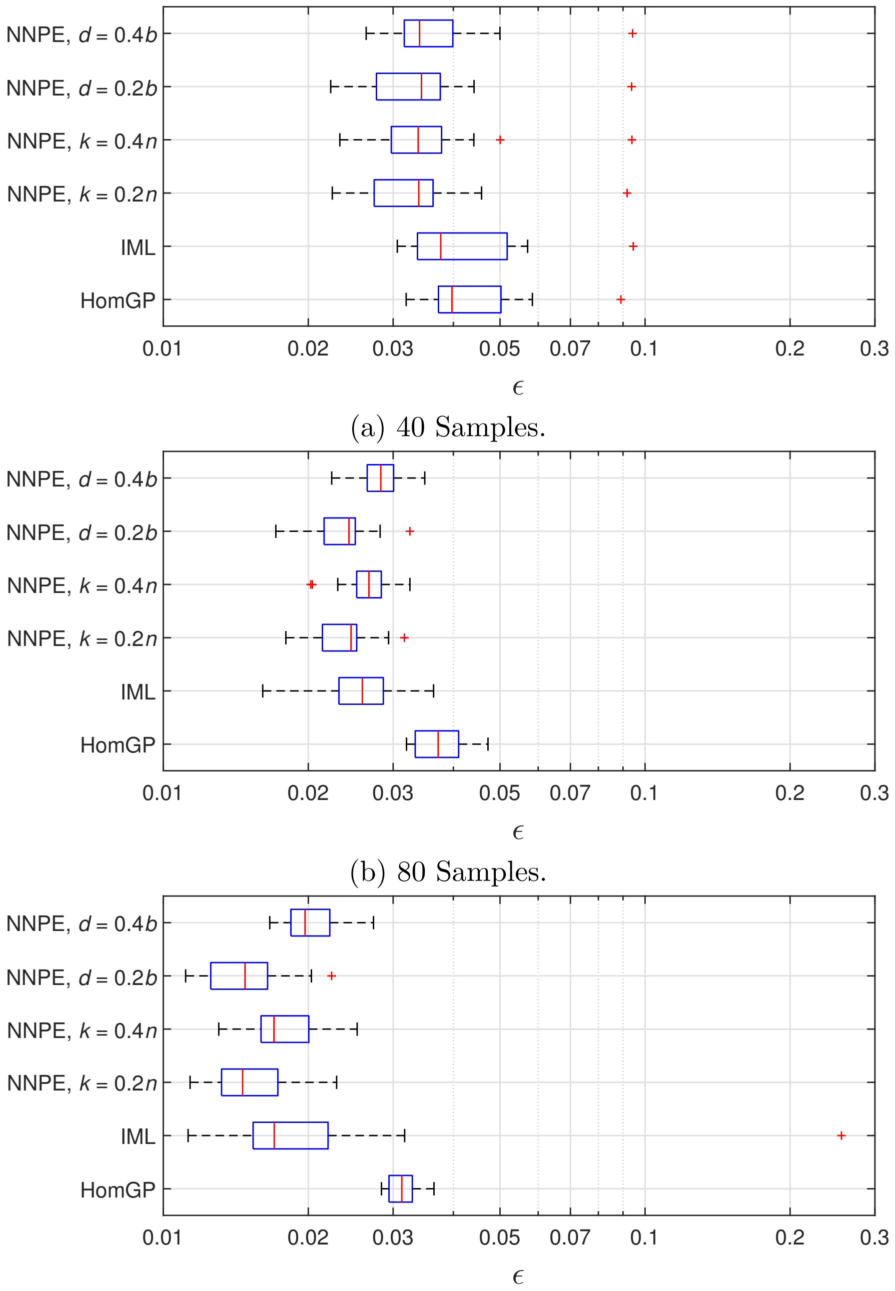
(b) NNPEHGP.

Figure IV.7: The worst IMLHGP and NNPEHGP ($k = 0.2n$) models in predicting the noise level for the two-dimensional mathematical function with $n = 80$. The transparent figures are the true noise standard deviation, while the filled figures are the predicted noise standard deviation.

however, IMLHGP can be observed to have a wider range of performances than NNPEHGP. The lack of robustness in the performances of IMLHGP can also be noticed by the appearance of a very poor outlier, and is even worse than the overall performances of standard homoscedastic GP. It is also interesting to obsevre that, for this problem, a smaller k and d value yields better NNPEHGP's predictive accuracy.

To further investigate the performances of IMLHGP and NNPEHGP in predicting the function's distributions, each best and worst results are also visualized in Figs. IV.9 and IV.10. The best and the worst here is defined according to the ε error metric. The results visualized here is when the models are trained using moderate number of training samples ($n = 80$). The $k = 0.2n$ configuration is chosen for the NNPEHGP as it achieved the best performance of NNPEHGP in this moderate training sample size.

With their best results, as shown in Fig. IV.9, IMLHGP and NNPEHGP shows similar predictions, that are successful in capturing the varying distribution of the function. A silight difference, however, still can be observed where IMLHGP is more precise in capturing the distributions within the low variance region. Similarly, the predictive distribution of NNPEHGP in the latter region has a slighly higher variance than the true distribution's variance. Whereas, in the high



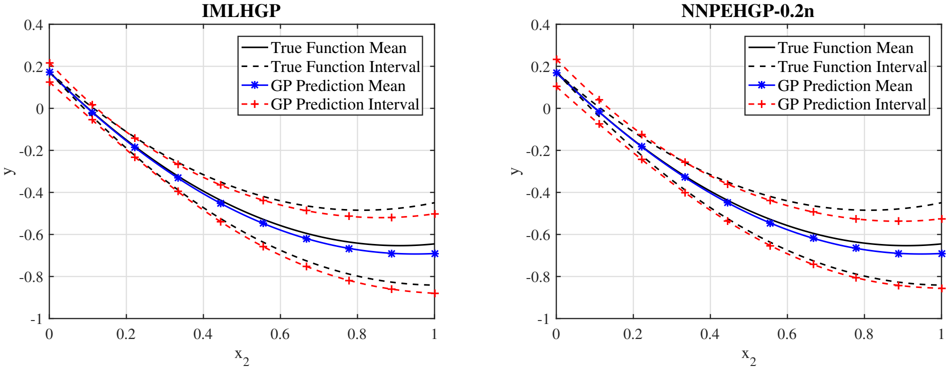
(a) 40 Samples.

(b) 80 Samples.

(c) 160 Samples.

Figure IV.8: Wasserstein distance results for two dimensional mathematical function.

variance region, NNPEHGP can be observed to have more accurate predictions. This phenomenon might be caused by the definition of the NNPEHGP model where the Bayesian residuals are smoothed first before being trained by a second GP model, which may cause the small Bayesian residuals to be transformed into higher values and likewise into lower values.

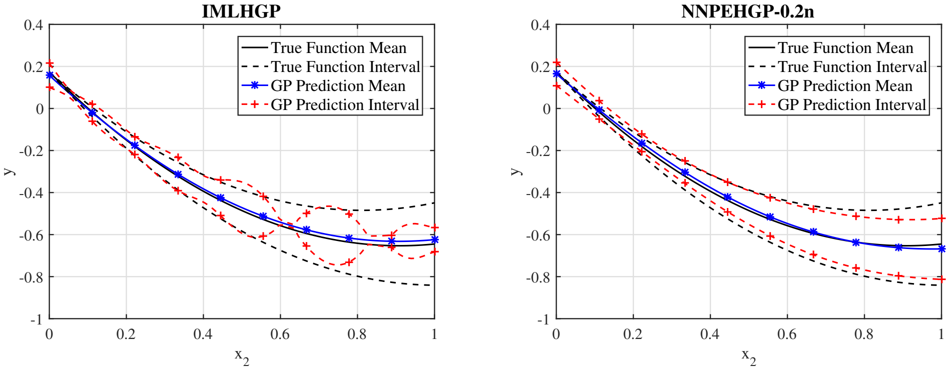


(a) IMLHGP.

(b) NNPEHGP.

Figure IV.9: The best IMLHGP and NNPEHGP ($k = 0.2n$) models in terms of the Wasserstein distance for the two-dimensional mathematical functions with $n = 80$ at $x_2 = 0.6$. The interval shows 95% error variation of the true function and the GP.

The difference between the two models here can be more clearly seen in their worst results, as shown in Fig. IV.10. IMLHGP can be observed to have a very poor prediction on the function distribution as it fails to capture the increasing trend of noise variance across the input spaces. The prediction of IMLHGP seems to severely interpolate the Bayesian residuals, especially in the high variance region where several predictions apprear close to zero. NNPEHGP model, at its worst, can be observed to still have accurate predictions on the function distribution, where it has been successful in capturing the varying distributions' variance across the input spaces. These visualizations further confirmed the superiority of NNPEHGP in terms of performance stability compared to IMLHGP when the number of training samples are small.



(a) IMLHGP.

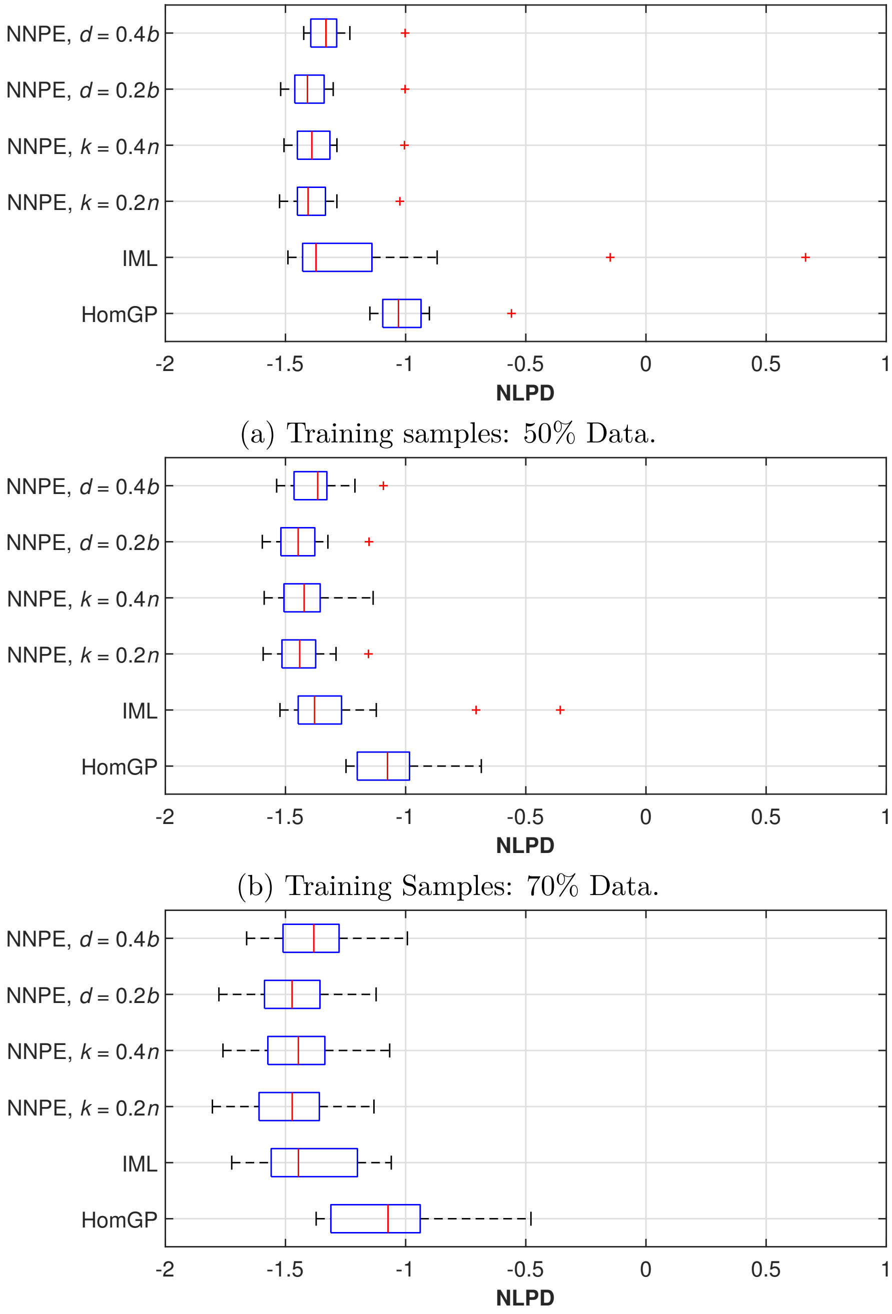
(b) NNPEHGP.

Figure IV.10: The worst IMLHGP and NNPEHGP ($k = 0.2n$) models in terms of the Wasserstein distance for the two-dimensional heteroscedastic function with $n = 80$ at $x_2 = 0.6$. The interval shows 95% error variation of the true function and the GP.

IV.2 Real-World Experimental Data

The results for the LIDAR experimental data are shown in Fig IV.11. For a low training sample size (Fig. IV.11a), it can be observed that every configuration of NNPEHGP performs similarly in predicting the distribution of test samples (in terms of NLPD) that shows significantly better performance compared to IMLHGP and standard homoscedastic GP. WHile IMLHGP also shoed a good performance in this low training sample size, the lack of robustness in its prediction can be noticed with its wide range of error accuracy and also the appearance of several terrible outliers.

A similar trend can also be observed for a moderate training sample size, as visualized in Fig. IV.11b. Here, it can be observed that IMLHGP's performances have improved from the previous low training sample size. However, several poor outliers can still be noticed in its predictive accuracy. Whereas NNPEHGP did not produce poor outliers as with the IMLHGP, which indicates that the former successfully avoids models that is wrongly interpolate the Bayesian residuals. When the number of samples use to train the models are large, it can be observed that the IMLHGP and NNPEHGP performs similarly in predicting the distribution of test samples (Fig. IV.11c) where both of the models yields a much better predictive accuracy than the standard homoscedastic GP. Yet, it still can also be



(a) Training samples: 50% Data.

(b) Training Samples: 70% Data.

(c) Training Samples: 90% Data.

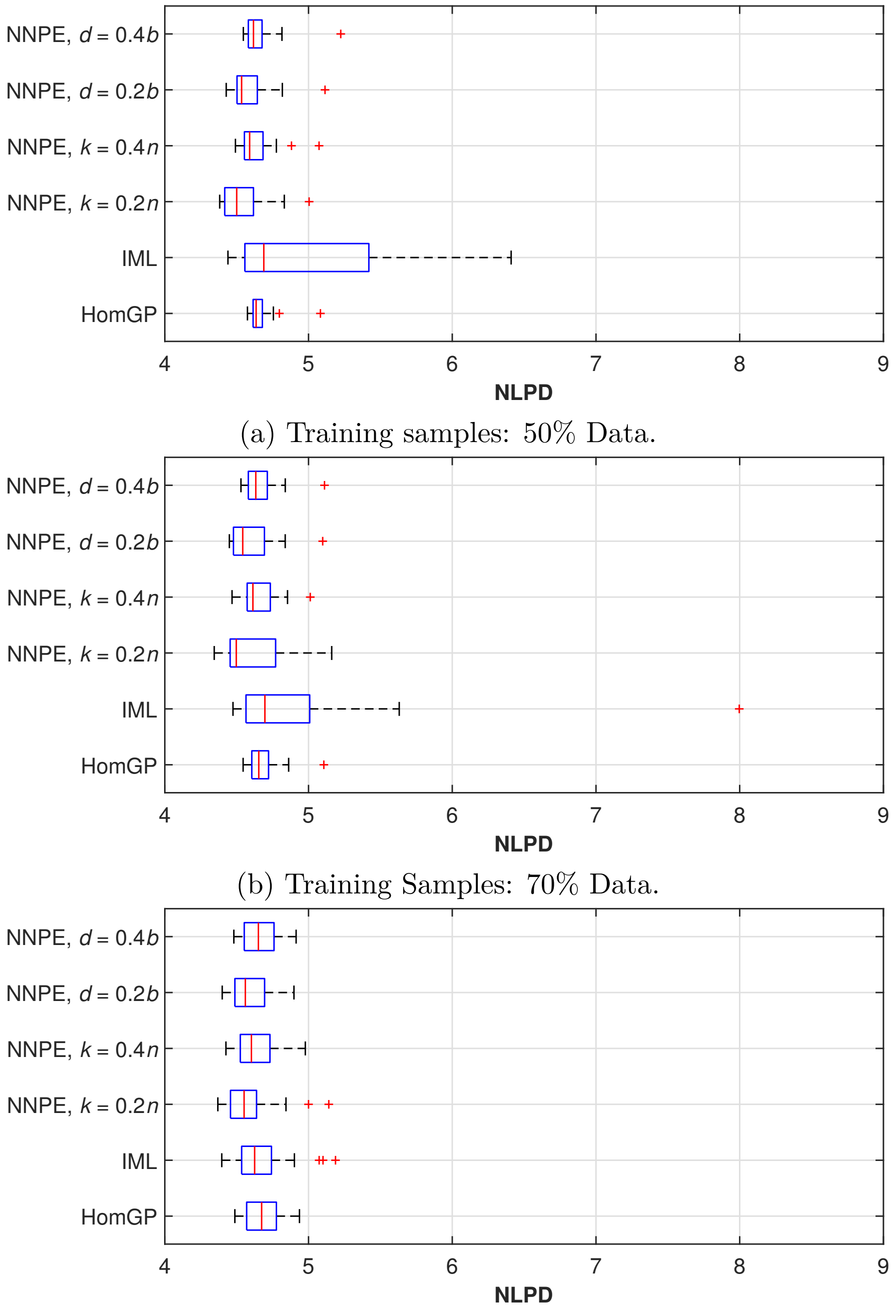
Figure IV.11: NLPD results for LIDAR experimental data.

seen that NNPEHGP with $k = 0.2n$ and $d = 0.2b$ slightly outperformed IMLHGP in predicting this experimental data when a larger number of training samples are available.

The performances of the model in predicting the distribution of motorcycle experimental data is shown in Fig IV.12. As explained in Section III.3, the distribution of the samples in this data is more complex than the previous LIDAR experimental data, yielding it more difficult for the model to learn the samples. In the low training sample size, as visualized in Fig. IV.12a, IMLHGP performed very poorly in predicting the distribution of the data even compared to the standard homoscedastic GP. NNPEHGP also does not provide any decent results, as the performaances are similar to that of the standard homoscedastic GP. This phenomenon confirms the complexity of the problem that this small size of training samples might be insufficient to train the models. However, the results of NNPEHGP can stil be seen to have a significantly better result than IMLHGP. This trend can also be observed in the moderate training sample size (Fig. IV.12b) and IMLHGP can be seen to have a terrible outlier in its predictive results.

The superiority of NNPEHGP can clearly be seen in the high training sample size, as presented in Fig. IV.12c. NNPEHGP yiels a better performance in predicting the distribution of the data compared to IMLHGP and standard homoscedastic GP, particularly when $k = 0.2n$ is used as the configuration for NNPEHGP. This result might not be interesting as the previous ones, as NNPEHGP only showed better performances for in this highest training sample size. However, the present training samples configuration still has a rather low size compared to the study in (Binois, Gramacy, and Ludkovski, 2018; Lázaro-Gredilla and Titsias, 2011; Zhang and Ni, 2020) where they also used replications as the training samples. The performance of the models in predicting this complex problem still confirmed a better predictive accuracy of NNPEHGP than IMLHGP.

To further investigate the performances of IMLHGP and NNPEHGP in predicting the distribution of the present complex problem, each of their best and worst predictions on learning the motorcycle experiment data with the moderate training sample size is visualized in Figs. IV.13 and IV.14. From their best results (Fig. IV.13), it can be observed that NNPEHGP is more precise in capturing the low variability in the left boundary region where IMLHGP still yields a high predictive variance. In the high variance region, NNPEHGP also has more accurate predictions. Though, IMLHGP gave a slightly more accurate prediction in the right boundary region.



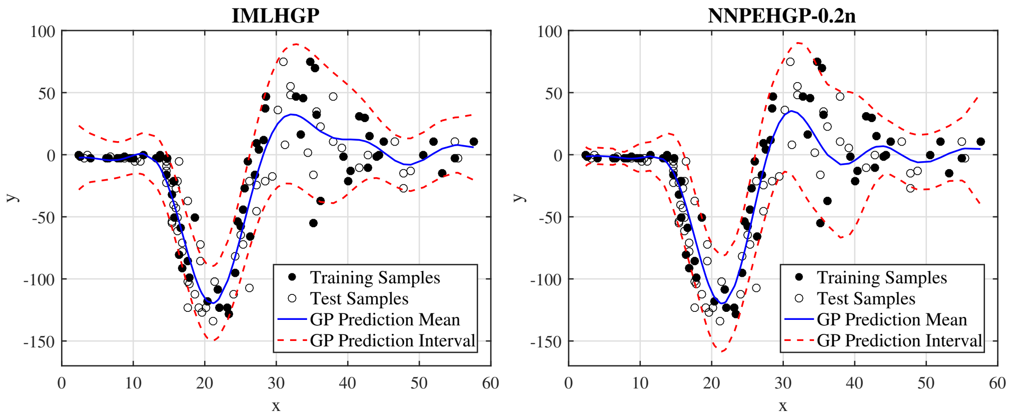
(a) Training samples: 50% Data.

(b) Training Samples: 70% Data.

(c) Training Samples: 90% Data.

Figure IV.12: NLPD results for motorcycle experimental data.

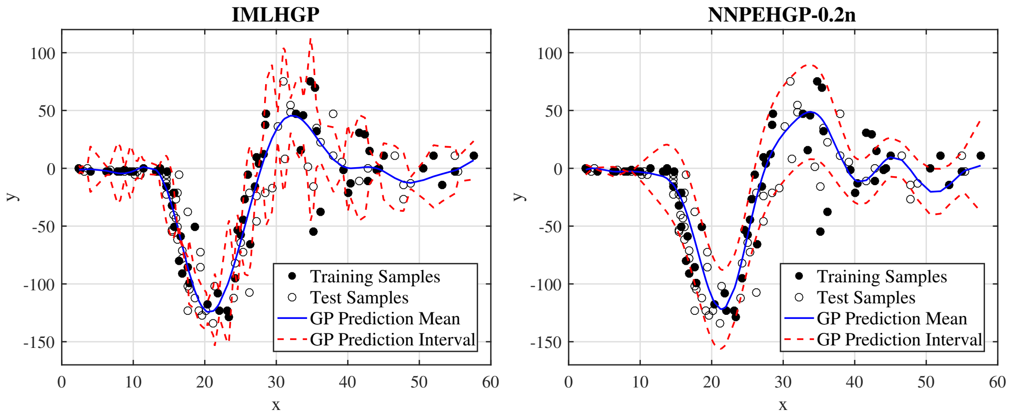
Focusing on the worst predictions, as presented in Fig. IV.14. The IMLHGP predictions is shown to lack robustness as IMLHGP seems to interpolate the scattered Bayesian residuals and thus yields terrible predictions in the distribution of the data. While NNPEHGP may have failed to accurately predict the high



(a) IMLHGP.

(b) NNPEHGP.

Figure IV.13: The best IMLHGP and NNPEHGP ($k = 0.2n$) models for motorcycle experiment data with moderate training sample size. The interval shows 95% error variation of the GP.



(a) IMLHGP.

(b) NNPEHGP.

Figure IV.14: The worst IMLHGP and NNPEHGP ($k = 0.2n$) models for the motorcycle experiment data with moderate training sample size. The interval shows 95% error variation of the GP.

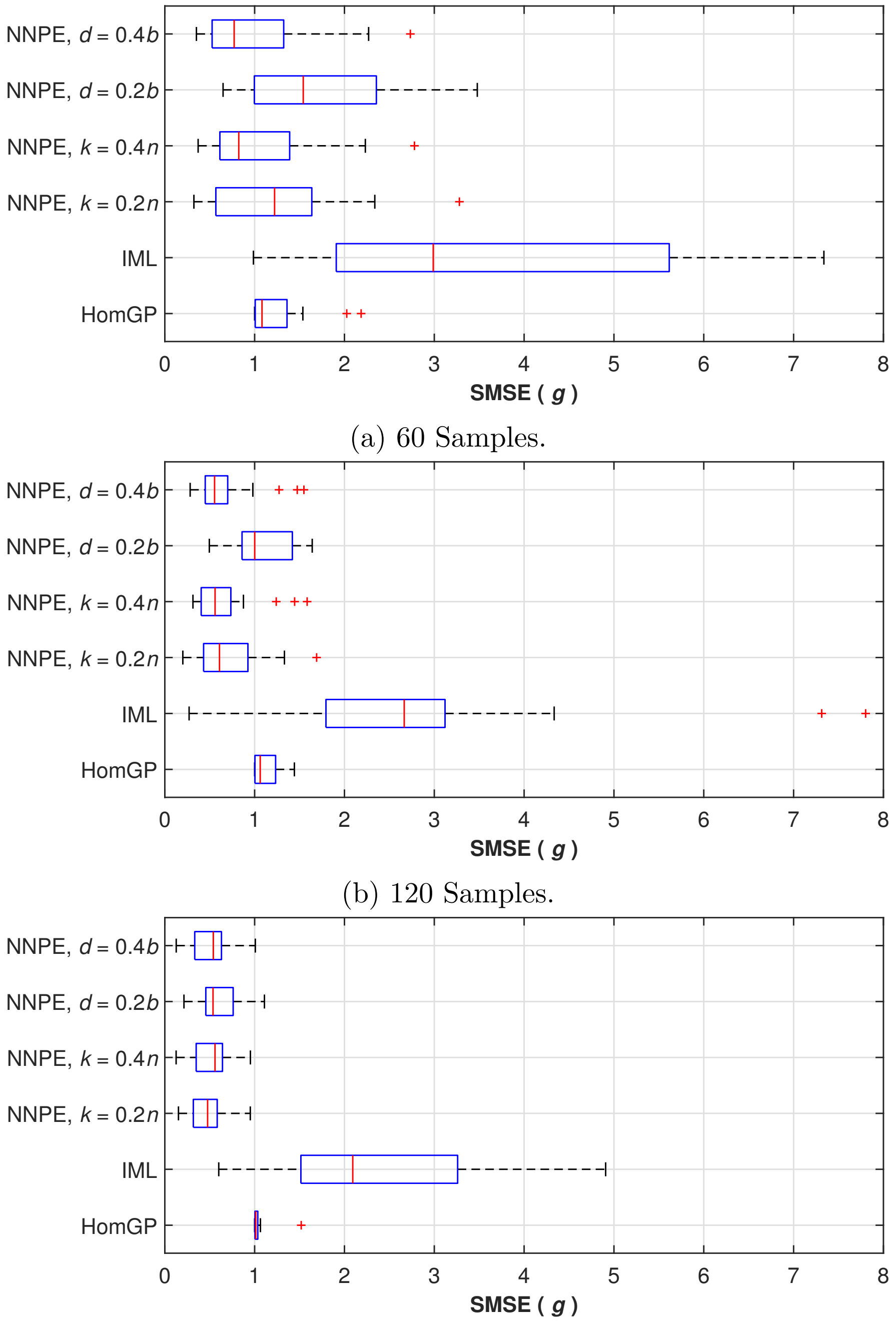
variance region, the model can still successfully capture the varying noise level across the input spaces. This result confirmed the superiority of NNPEHGP to IMLHGP in the stability of the predictions.

IV.3 Real-World Analytical Problems

IV.3.1 Three-Dimensional Operational Envelope Analysis of A Generic Transporation Aircraft Under Uncertainty

The performances in predicting the noise standard deviation of this three dimensional analytical problem is presented in Fig. IV.15. In the low training sample size ($n = 60$), it can be observed the predictive performance of IMLHGP is very poor as the median is a lot higher than the standard homoscedastic GP, its performances are also showed to be really unstable. The performances of NNPEHGP can also be observed to be not really decent in this low training sample size. Every NNPEHGP configuration yielded a not robust performances in predicting the noise level trend. Observing the good performances of standard homoscedastic GP, it might seem that the noise level is not that variative in the input spaces, and the low training sample size used to train the models might yielded to a poor predictions in this high dimension problem. Nonetheless, it still can be observed that NNPEHGP still performed much better than IMLHGP. Particularly for NNPEHGP that uses $k = 0.4n$ and $d = 0.4b$, its median are also lower than the standard homoscedastic GP, indicates the performances of NNPEHGP in general are also better than the standard homoscedastic GP.

The superiority of NNPEHGP in predicting the noise level in this problem can be more observed in the moderate and high training sample size ($n = 120$ and $n = 240$ in Figs. IV.15b and IV.15c), respectively). It can be observed that most of the results given by NNPEHGP is significantly better than IMLHGP and standard homoscedastic GP, while IMLHGP still yielded poor and unstable predictive accuracy. However, it can also be observed that NNPEHGP with $d = 0.2b$ showed rather poor performances, even compared to the standard homoscedastic GP. This might indicates that in this higher dimensional problem with a rather low variability in the noise level, more number of nearest neighbors results in a better prediction for the noise level as NNPEHGP with $k = 0.4n$ and $d = 0.4b$ showed better results. It can also be inferred that the configurations for NNPEHGP is more stable when the number of nearest neighbors is used rather than the



(a) 60 Samples.

(b) 120 Samples.

(c) 240 Samples.

Figure IV.15: SMSE (g) results for three-dimensional operational envelope problem.

distance of its nearest neighbors.

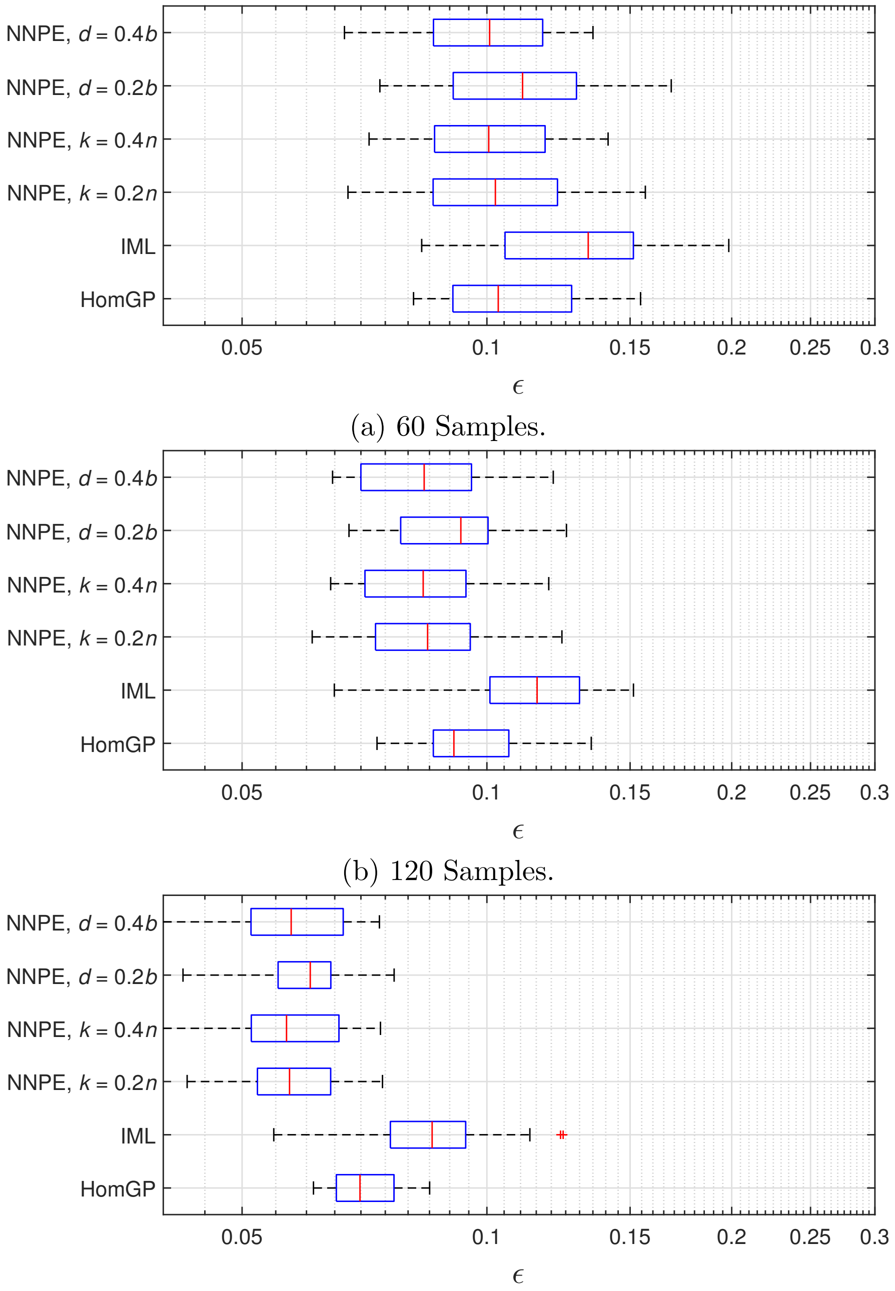
In the high training sample size, it can be observed that IMLHGP still yielded poor predictions while each configuration of NNPEHGP showed similarly fine predictions. This result also showed the poor predictive accuracy of IMLHGP in learning high dimensional problems. It can also be observed that the $k = 0.2n$ configuration for NNPEHGP showed slightly better performances than the other configurations. This result might indicate that a smaller k parameter for NNPEHGP is preferred when the number of training samples are high.

The results of the models in predicting the distributions are showed in Fig. IV.16. It can be observed in the low training sample size (Fig. IV.16a), the predictive performance of IMLHGP seemed to be poor, even worse than the standard homoscedastic GP. Whereas NNPEHGP yielded a better predictive accuracy than IMLHGP, particularly NNPEHGP with $k = 0.2n$, $k = 0.4n$, and $d = 0.2b$. Even the performances of NNPEHGP may seemed to be similar to the standard homoscedastic GP, NNPEHGP still can achieve more accurate predictions.

A similar trend can also be observed in the moderate training sample size (Fig. IV.16b, where IMLHGP still yielded poor performances in predicting the distribution of this analytical problem. Contrast to NNPEHGP, where the model has already yielded more accurate predictions compared to the standard homoscedastic GP and IMLHGP. The total range in the performances of NNPEHGP can also be observed to be lower than IMLHGP and standard homoscedastic GP's performance range; showed the robustness of NNPEHGP.

The performances of NNPEHGP is much improved in the high training sample size, as presented in Fig. IV.16c. Each configuration of NNPEHGP can be clearly observed to gave significantly better predictions than IMLHGP and standard homoscedastic GP. Its median is much lower and the worst predictions is also showed to be not severely bad. In this high training sample size, the performances of IMLHGP are still showed to be poor, that its performances are worse than the standard homoscedastic GP. This results can affirm that the natural procedure of IMLHGP might be not enough to model a high dimensional heteroscedastic problem, as high training sample size is still insufficient to give fine predictions.

In this high dimensional problem, it is also interesting to see the visualization of the predicted distributions, presented in Fig. IV.17. Probability density functions (PDF) at two distinct input locations are visualized here. The first example is



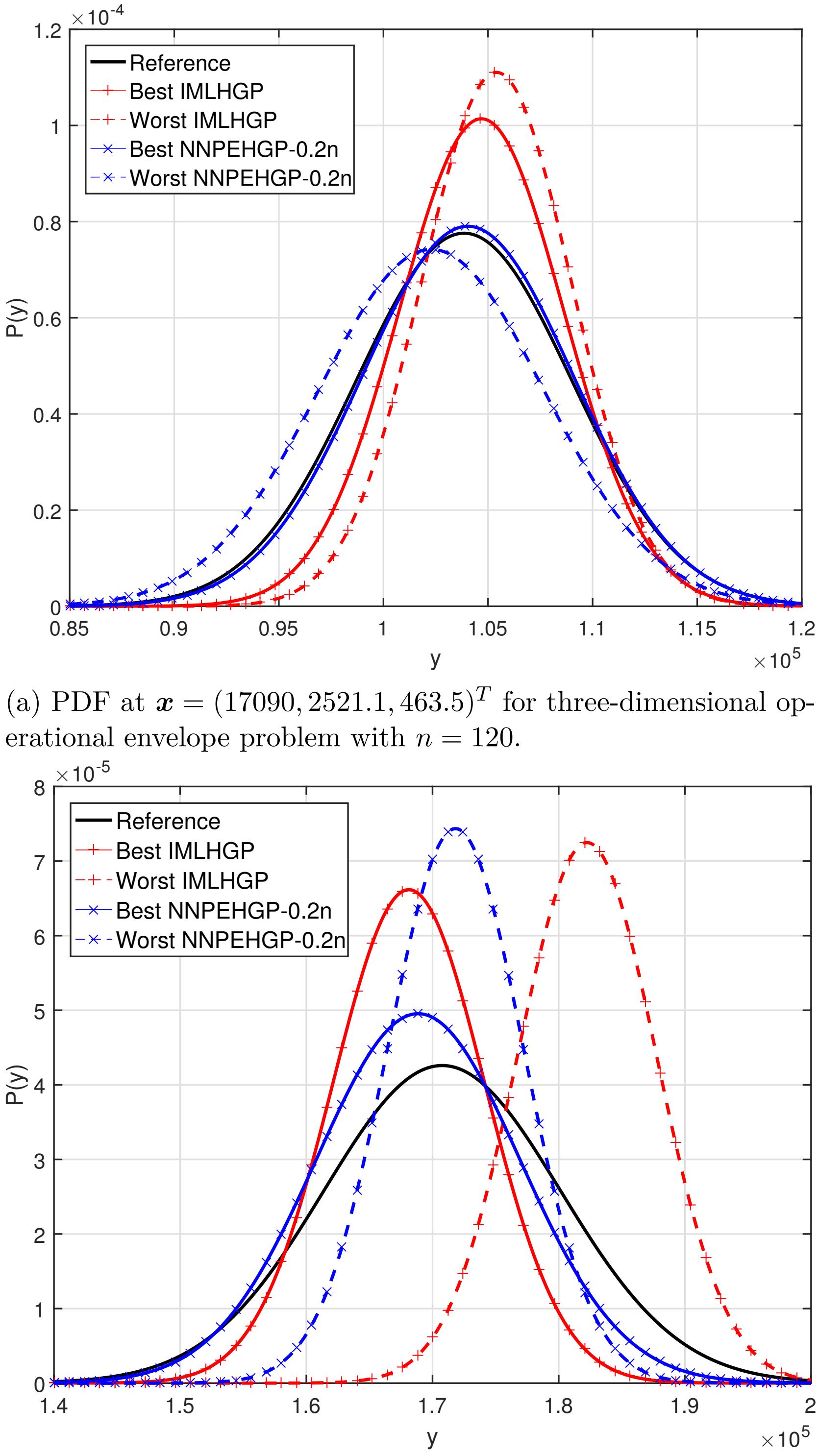
(a) 60 Samples.

(b) 120 Samples.

(c) 240 Samples.

Figure IV.16: Wasserstein distance results for three-dimensional operational envelope problem.

visualized at the location where the expected fuel weight and its variance is rather small, presented in Fig. IV.17a. The next example is visualized at the location where the expected fuel weight is the highest in the input spaces, that also found



(a) PDF at $\boldsymbol{x} = (17090, 2521.1, 463.5)^T$ for three-dimensional operational envelope problem with $n = 120$.

(b) PDF at $\boldsymbol{x} = (153810, 2785, 463.5)^T$ for three-dimensional operational envelope problem with $n = 120$.

Figure IV.17: Comparison of the predicted PDF for three dimensional operational envelope problem with $n = 80$.

to have the highest variability as well, shown in Fig. IV.17b. Here, the best and worst prediction of IMLHGP and NNPEHGP in moderate training sample size is visualized along with the reference distributions. This reference distributions is obtained by doing 10^5 realizations at the inputs. The $k = 0.2n$ configuration for NNPEHGP is chosen here as it shows the best performance of NNPEHGP in the moderate training sample size.

Observing the results in the first example (Fig. IV.17a), it can be observed that the prediction of NNPEHGP in its best results predicted really well the distribution of the fuel weights, as its PDF looked very similar to the reference one. In its worst result, NNPEHGP can also be observed to still gave a fine predictions. Whereas IMLHGP is observed to perform poorly in predicting the distribution, both at its best and worst results. Although, the model successfully captured the mean of the distributions, it fails to capture the variance of the full weight as its predicted variance seemed to be smaller than the reference distribution.

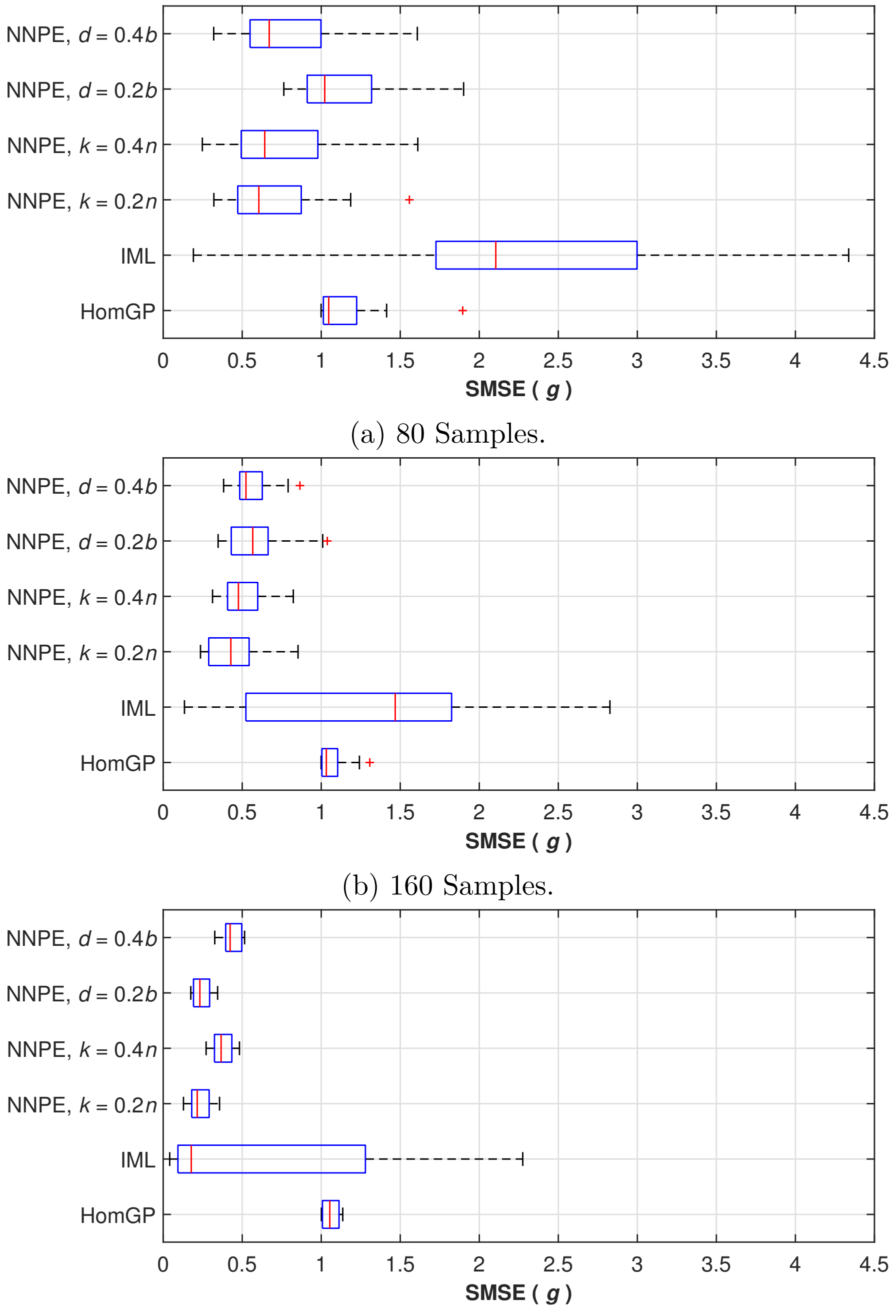
Continued by observing the second example (Fig. IV.17b) where the variance of the fuel weight is higher. In the study of design exploration using this analytical problem, this specific input location seemed to be important as it has the highest fuel weight and variance that the designer would need to know. The high variance in this example definitely makes the model a lot more difficult in predicting distribution, as the observed sample can be really scattered. In this example, it can be observed that NNPEHGP at its best result can predict the reference distribution quite accurately while IMLHGP at its best result seemed to only give fine prediction on the mean. It can be seen that the predicted variance of IMLHGP is lower than the reference distribution. This result, along with the previous example, might indicates the wrongly interpolating phenomena that appear in IMLHGP predictions on the noise level while NNPEHGP successfuly avoids them. In this second example, IMLHGP and NNPEHGP, at its worst condition are observed to gave poor prediction on the variance of the fuel weights. This might be caused by the insufficient sampling around the region to give informations in training the model. However, NNPEHGP at its worst result can be observed to still gave fine prediction on the mean of distributions while IMLHGP yielded a very poor prediction.

IV.3.2 Four-Dimensional Temperature Modeling of A Solid Sphere Immersed in Fluid under Uncertainty

The results for this four-dimensional analytical problem in terms of SMSE (g) are shown in Fig IV.18. In the low training sample size ($n = 80$), it can be observed from Fig. IV.18a that IMLHGP yielded very poor predictions on the noise level, even much worse than the standard homoscedastic GP. Though IMLHGP may found one of the best prediction, the general performances can be clearly to observed to be really not robust. The median of IMLHGP's performances is also appeared to be higher than the other models. NNPEHGP, particularly with $k = 0.2n$, $k = 0.4n$, and $d = 0.4b$, is already yielded fine noise level predictions that is significantly more accurate than IMLHGP. However, it can also be seen that NNPEHGP with $d = 0.2b$ is not giving a fine prediction on the noise level as its performances can be observed to be slightly worse than the standard homoscedastic GP. As pointed in the previous example, the number of nearest neighbor configuration might be preferred in training NNPEHGP model as its performances are seemed to be more robust. Still, more experiments are needed to investigate this specific detail of NNPEHGP.

In the moderate training sample size ($n = 160$), as presented in Fig. IV.18b, it can be observed that every configuration for NNPEHGP already yielded fine and robust predictions on the noise level while IMLHGP still gave poor and unstable predictions. The performances of IMLHGP in this moderate training sample size can still be observed to be worse than the standard homoscedastic GP. As found in the previous high dimension problem, IMLHGP seems to need a really high number of training samples to give fine predictions on the noise level. It can also be seen that NNPEHGP based on the number of nearest neighbors still gave slighltly finer prediction than the configuration based on the maximum distances to its neighbors.

The performance of IMLHGP in predicting the noise level can be observed to be much improved in the high training sample size ($n = 320$), visualized in Fig. IV.18c, as it gives the lowest median than ones achieved by other models. However, its lack of robustness can be still clearly observed, that several results appeared to be worse than the overall results of standard homoscedastic GP. Here, the perfor-



(a) 80 Samples.

(b) 160 Samples.

(c) 320 Samples.

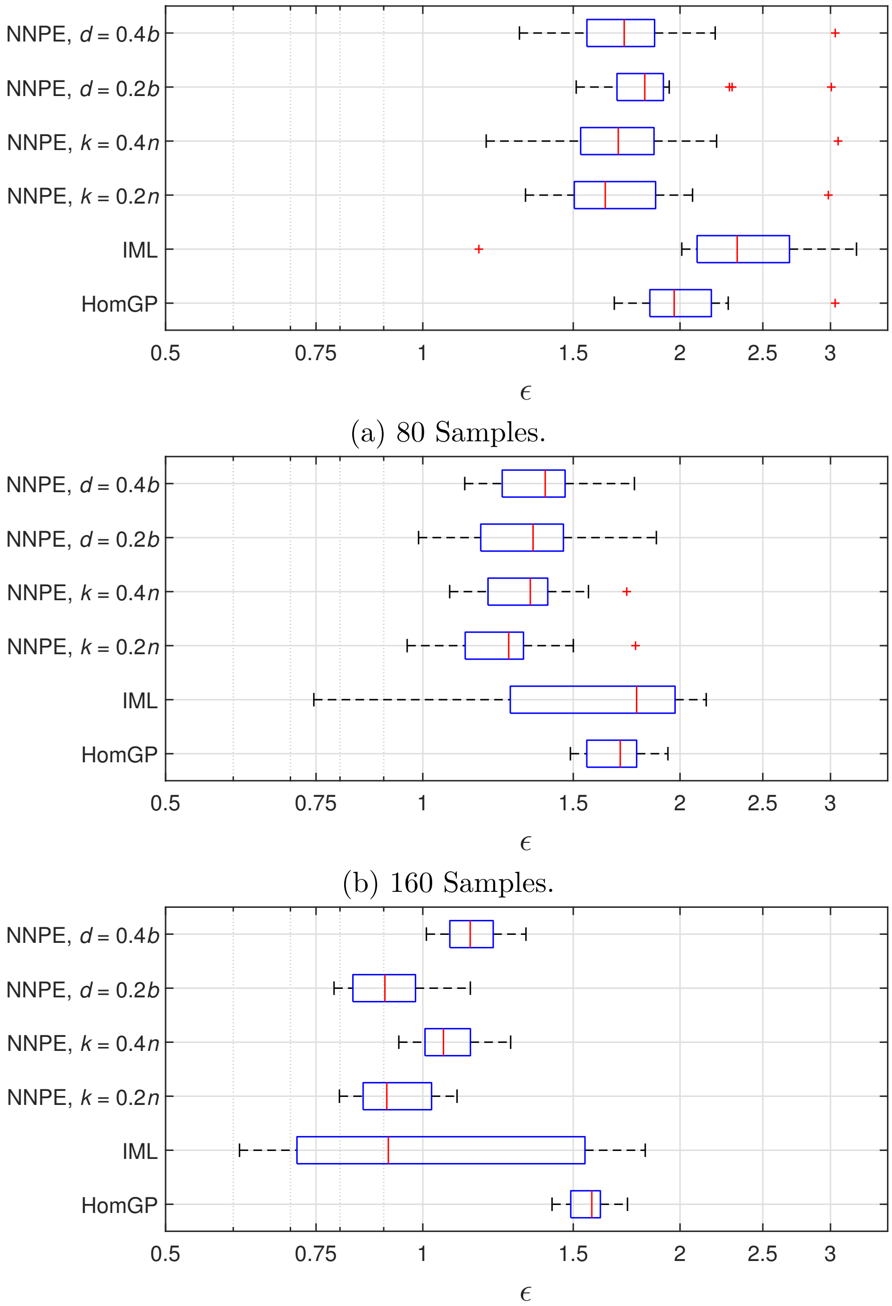
Figure IV.18: SMSE (g) results for four-dimensional temperature modeling problem.

mances of NNPEHGP can be seen to be really fine in predictiong the noise level, indicated by its low median and the robust performances. Though the median of NNPEHGP is not lower than IMLHGP, the discrepancy is not significant while the high robustness of NNPEHGP compared to IMLHGP can be clearly noticed. This result still confirms the superiority of NNPEHGP in learning varying noise level of a high dimensional problem. It can also be observed that the smaller parameter for NNPEHGP's configuration, i.e., $k = 0.2n$ and $d = 0.2b$, is more preferable when high number of samples is available in training the model as they showed the best results in this high training sample size. When a lot information of Bayesian residuals is available, this smaller parameter would make the model capture the noise level in each region more precisely.

The better predictive accuracy and robustness of NNPEHGP can also be seen in terms of the predictive posterior distributions, as shown in Fig. IV.19. In the low training sample size, as presented in Fig. IV.19a, it can be observed NNPEHGP outperforms other model with a significantly better predictive accuracy than IMLHGP and standard homoscedastic GP. IMLHGP might obtain the best predictive accuracy; however, it can be observed the performances are very not robust since the best performance itself is an outlier. IMLHGP seemed to have poor predictive performance in low training sample size as its performances can be observed to be generally worse than the standard homoscedastic GP.

In the moderate training sample size (Fig. IV.19b), it can be observed that IMLHGP still achieved one of the best prediction on the distribution of the temperature; however, its lack of robustness can be still clearly observed. Clearly, this indicates that IMLHGP might perform well in the case of lucky samples, but its performance is significantly deteriorates for an unlucky sampling plan. Here, it can be observed that NNPEHGP yielded the finest median value than the other models and also a more robust performances than IMLHGP.

The performances of IMLHGP is much improved in the high training sample size (Fig. IV.19c) as it yielded low median value and also achieved one of the best predictions. However, it can be observed that the performances is still not robust as the range of its error is still wide. It can also be observed that NNPEHGP still gives more stable predictions than IMLHGP, with the median of ε is similar to that of the IMLHGP. This observations also supports the hypothesis that IMLHGP performs well when the number of training samples is quite high but relatively poor on a low training sample size. This also confirms the superiority



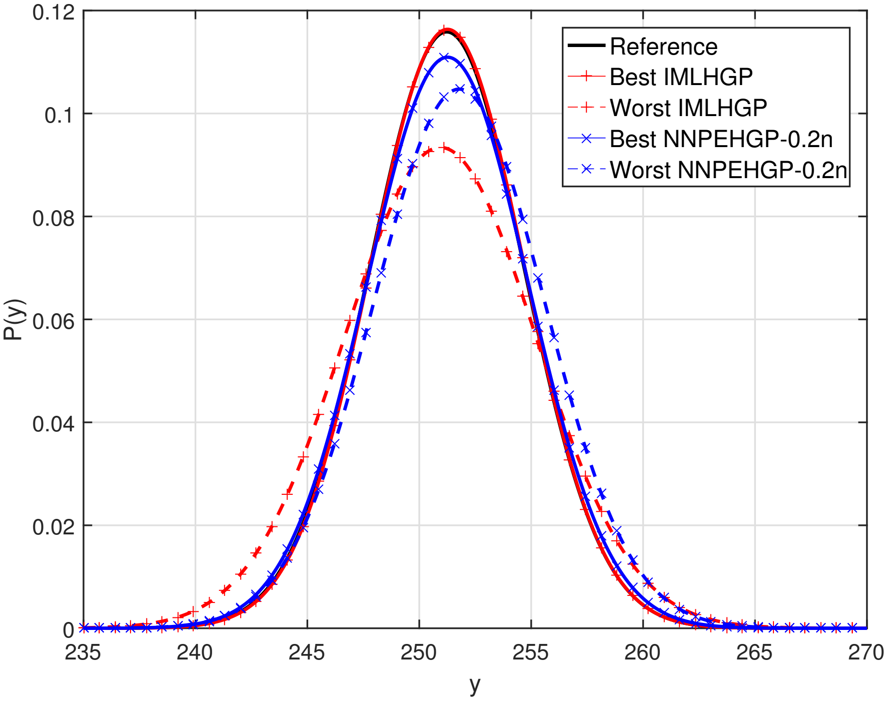
(a) 80 Samples.

(b) 160 Samples.

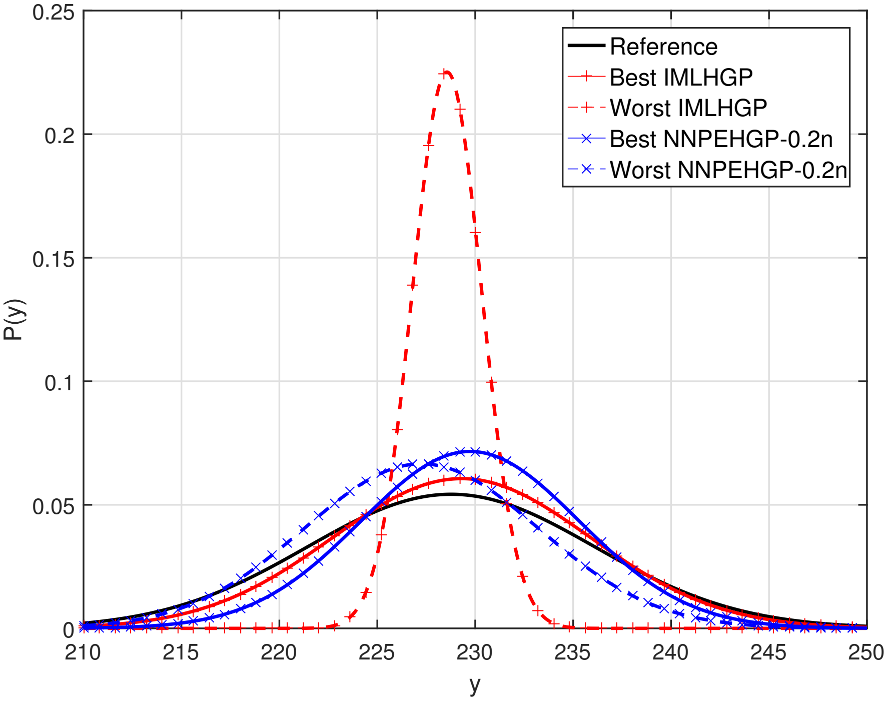
(c) 320 Samples.

Figure IV.19: Wasserstein distance results for four-dimensional temperature modeling problem.

of NNPEHGP as it still gives more robust predictions than IMHGP even though several results provided by IMLHGP showed higher accuracy than NNPEHGP in the high training sample size.



(a) PDF for $\boldsymbol{x} = (0.6965, 0.6955, 0.7225, 0.4225)^T$ for four-dimensional temperature modeling problem with $n = 160$.



(b) PDF for $\boldsymbol{x} = (0.4215, 0.2125, 0.0195, 0.6615)^T$ for four-dimensional temperature modeling problem with $n = 160$.

Figure IV.20: Comparison of the predicted PDF for four-dimensional temperature modeling problem.

As the previous three-dimensional problem, the predicted distribution for this four-dimensional temperature modeling problem is also visualized in its PDF form, presented in Fig. IV.20. The PDFs are visualized in two distinct input locations, where the first PDF represents the temperature distribution which has low variance while the second PDF represents a higher variance, shown in Figs. IV.20a and Fig. IV.20b respectively. The second example also found to be one of the locations where the expected temperature is low that might be important in the application of this empirical problem. The best prediction of IMLHGP and NNPEHGP along with the worst prediction in moderate training sample size are visualized here. The analysis will be conducted by comparing the predictions to the reference distribution. The reference distribution in this problem is already obtained by doing 10^4 replications at each inputs. The $k = 0.2n$ configuration for NNPEHGP is chosen in this example as this configuration represents the best performance of NNPEHGP.

In the first example (Fig. IV.20a), it can be observed that IMLHGP and NNPEHGP performed similarly fine in predicting the distribution at its best result, with IMLHGP slightly outperformed the latter. Observing the performances of the two models at each of their worst result, it can be seen that the prediction of IMLHGP is rather poor than NNPEHGP as the predicted variance of the former has a slightly higher difference to the reference. Nonetheless, the performances of both models can be observed to be not that different, showed the similarity in the accuracy of the two models when the location has a rather low variability in the samples.

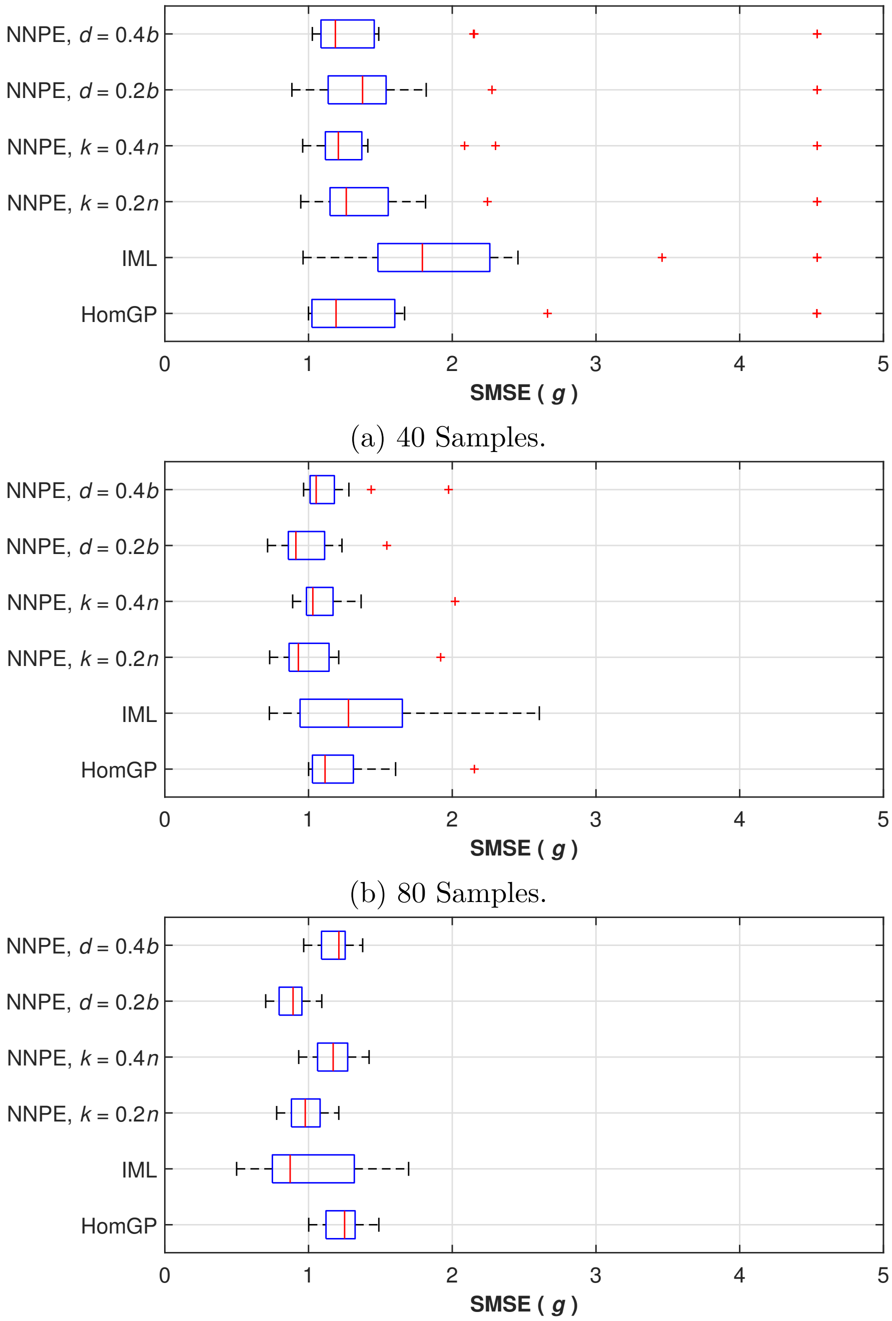
Continued by observing the second example (Fig. IV.20b), a distinct performances can be seen between IMLHGP and NNPEHGP. In this high variance location, NNPEHGP at its best and worst result predicted the reference distribution quite accurately, although NNPEHGP at its worst result yielded a slightly inaccurate prediction on the expected temperature. In its best result, IMLHGP might gave a slightly better prediction than NNPEHGP as its expected variance is more precise. However, IMLHGP at its worst result yielded a terrible prediction on the variance of the temperature; that might be caused by the wrongly interpolating phenomena that often appears in the IMLHGP predictions. This result also confirms the superiority of NNPEHGP in the robustness of the predictions.

IV.4 Stochastic Simulator

The performances in predicting the noise standard deviation for this stochastic simulator problem is presented in Fig. IV.21. The complexity of this problem can be investigated by observing the performances in the low training sample size ($n = 40$) results, shown in Fig. IV.21a. It can be observed that IMLHGP performed poorly in predicting the noise level when the number of samples to train the model are quite low. The performances of IMLHGP can be seen as significantly worse than the standard homoscedastic GP. NNPEHGP also not performed really fine in capturing the noise level as its performances are rather similar to the standard homoscedastic GP. Here, it can be noticed the complexity of the samples generated by this stochastic simulator, 20 number of samples may not be sufficient in training the models. Nonetheless, NNPEHGP with $k = 0.4n$ and $d = 0.4b$ for its configuration showed slightly better results compared to the standard homoscedastic GP. NNPEHGP also can be seen to significantly outperfroms IMLHGP in this low training sample size.

Although the performances of IMLHGP can be observed to be much improved in the moderate training sample size ($n = 80$), it can be seen from Fig. IV.21b that the model still gave unstable predictions and its general performances are still worse than the standard homoscedastic GP. In this moderate training sample size, NNPEHGP can be seen as the best model in predicting the noise level of the samples, especially NNPEHGP with $k = 0.2n$ and $d = 0.2b$. Even though NNPEHGP could not avoid outliers that are also appeared in every model, it can be observed that the model generally have better and more robust performances than the other models.

The superiority of NNPEHGP model can also be observed in the high training sample size ($n = 160$), as visualized in Fig. IV.21c. NNPEHGP can be noticed to have more stable performances with the increasing number of training samples while IMLHGP still showed its lack of robustness. In this high training sample size, IMLHGP can be observed to achieve the best performance median that shows its high requirement on the size of the samples to train the model. Nevertheless, the NNPEHGP model might still be more preferable in learning this stochastic simulator as its performances are significantly more robust than IMLHGP. Still,



(a) 40 Samples.

(b) 80 Samples.

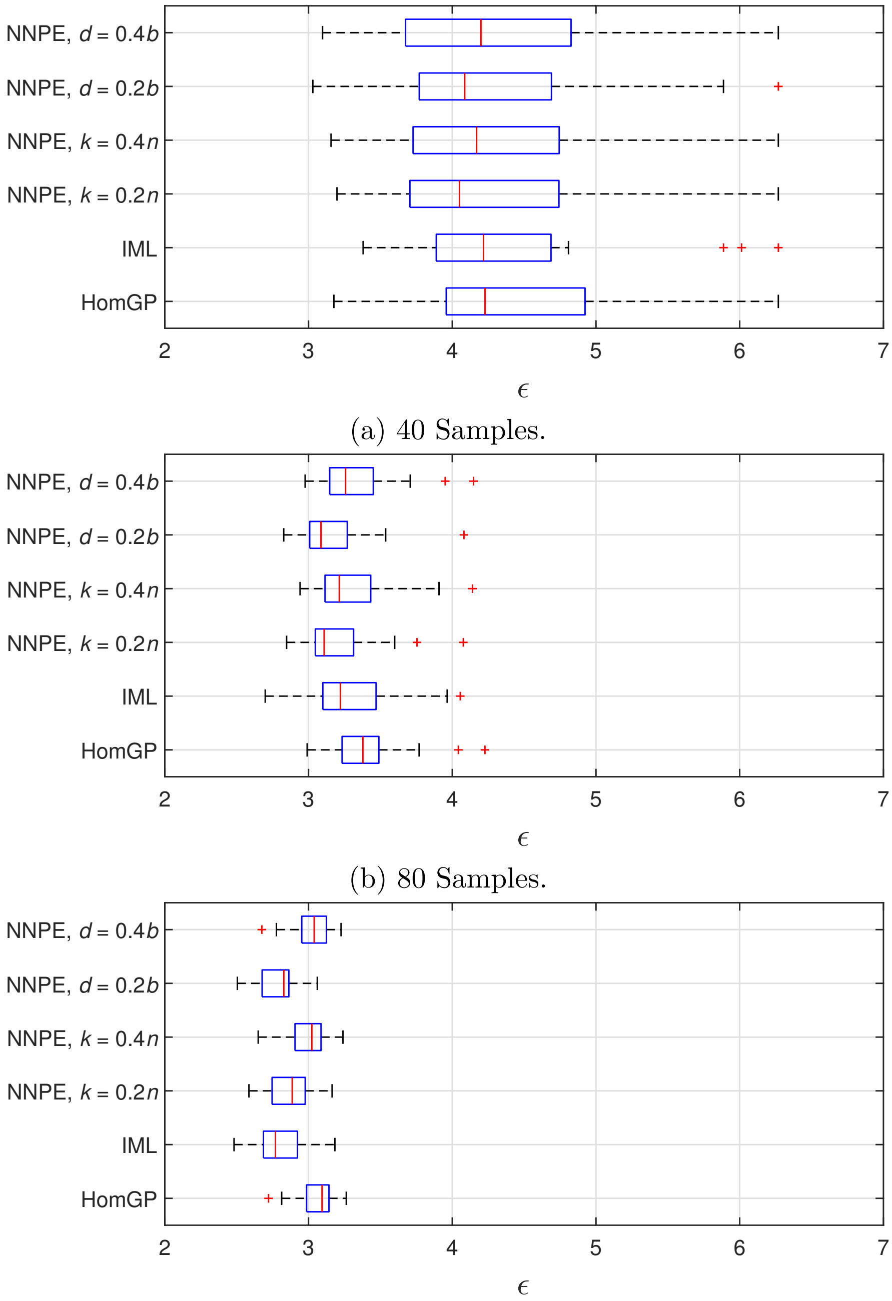
(c) 160 Samples.

Figure IV.21: SMSE (g) results for the stochastic simulator problem.

the parameters for NNPEHGP configuration should be chosen carefully, as the higher parameters, i.e., $k = 0.4n$ and $d = 0.4b$, showed rather poor results that can be seen to have similar results to that of standard homoscedastic GP. As pointed in the previous examples, a smaller parameter for NNPEHGP model might be preferred when the number of samples to train the model is high.

The performance of the models in predicting the distribution over the samples of this stochastic simulator is presented in Fig. IV.22. As pointed in Section III.5, the distribution of the samples generated by the simulator is not generally Gaussian, that could not capture accurately by the assumption in GP models used in this study. It can be observed that IMLHGP and NNPEHGP performed quite similarly at each training sample size. Though both models have similar results to the standard homoscedastic GP in the low training sample size, they can be noticed to outperform the standard homoscedastic GP when the number of samples to train the model are increased.

Although NNPEHGP did not significantly outperforms IMLHGP in this example, the proposed model at least have similar results to the benchmark IMLHGP model. These results showed the inability of NNPEHGP and IMLHGP in capturing the non-Gaussian distribution as the models used the assumption of Gaussian distrbuted noise in defining the surrogate models. More studies then need to be conducted in further developing the models to capture non-Gaussian distributions of a function. Nevertheless, NNPEHGP still seemed to be promising as the model has similar performances to IMLHGP in capturing the total distribution and significantly outperformed IMLHGP in predicting the noise level.



(a) 40 Samples.

(b) 80 Samples.

(c) 160 Samples.

Figure IV.22: Wasserstein distance results for the stochastic simulator problem.

Chapter V

Conclusion and Future Works

V.1 Conclusions

In this thesis, a Nearest Neighbor Point Estimates HGP (NNPEHGP) model is proposed to approximate black-box functions with heteroscedastic noise. This proposed model follows the main idea in MLHGP and IMLHGP by using a point estimate to replace the full noise posterior distribution. In contrast to IMLHGP, NNPEHGP uses an extra distance-based non-parametric regression to reduce the wrongly interpolating phenomenon that frequently appears in IMLHGP noise level prediction, especially when the training sample size is low. The proposed method works by performing a smoothing pre-step to the Bayesian residual before it enters the GP formulation. This modification is showed to significantly improve the accuracy of the HGP model on learning functions with heteroscedastic noise.

To validate the accuracy of the proposed NNPEHGP model, its performance is compared with the standard homoscedastic GP and the IMLHGP model on two heteroscedastic mathematical functions, two real-world experimental datasets, two real-world analytical problems under uncertainties, and a stochastic simulator. The results show that the proposed NNPEHGP outperforms both the homoscedastic GP and the IMLHGP in accuracy and stability, especially on a low training sample size. NNPEHGP also performs very well with high training sample size, highlighting the high robustness of the model. The results also show that NNPEHGP is giving much more stable predictions in high dimensional problem. The extra computational cost of NNPEHGP is also very cheap since the extra non-parametric regression is computationally simple.

In this study, two configuration for the extra non-parametric regression in NNPE-

HGP procedure is used; based on the number of nearest neighbors and on the maximum distances. Although more experiments are still needed, the results show that NNPEHGP which uses the number of nearest neighbors gave finer predictions than the other configuration. The results also show, that in general, a smaller number of nearest neighbors is preferable when the training sample size is high, and vice versa, a higher number of nearest neighbors when the training sample size is rather low.

V.2 Future Works

For future works, it is suggested to explore the possibility of automatic optimization of the parameters, i.e., k and d in the distance-based smoothing regression procedure as this might improves the performance of NNPEHGP furthermore. This could also make NNPEHGP model to be more adaptive to the available number of training samples. It is also interesting to explore the possibility of using different k or d values at each training by adapting to the observed samples. In several complex heteroscedastic problems, the utilization of replication might also still recommended to capture a more precise distribution over the samples. Thus, another potential research is to extend the concept of NNPEHGP to heteroscedastic problems with sample replications, such as in the Stochastic Kriging (Ankenman, Nelson, and Staum, 2008) and Practical Heteroscedastic GP method (Binois, Gramacy, and Ludkovski, 2018). Finally, observing the performances of NNPEHGP that is rather poor in learning the non-Gaussian distribution, it will also be interesting to extend the concept developed in this thesis to solve problems with non-Gaussian noise (Wang, 2014).

References

Andrés, E. et al. (2012). "Efficient aerodynamic design through evolutionary programming and support vector regression algorithms". In: *Expert Systems with Applications* 39.12, pp. 10700 –10708. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.02.197. URL: http://www.sciencedirect.com/science/article/pii/S0957417412004708.

Ankenman, Bruce, Barry L Nelson, and Jeremy Staum (2008). "Stochastic kriging for simulation metamodeling". In: *2008 Winter Simulation Conference*. IEEE, pp. 362–370.

Bae, Harok, Daniel L Clark, and Edwin E Forster (2019). "Nondeterministic Kriging for Engineering Design Exploration". In: *AIAA Journal* 57.4, pp. 1659–1670.

Bhattrai, Sudip, Jouke HS de Baar, and Andrew J Neely (2018). "Efficient uncertainty quantification for a hypersonic trailing-edge flap, using gradient-enhanced kriging". In: *Aerospace Science and Technology* 80, pp. 261–268.

Binois, Mickael, Robert B Gramacy, and Mike Ludkovski (2018). "Practical heteroscedastic gaussian process modeling for large simulation experiments". In: *Journal of Computational and Graphical Statistics* 27.4, pp. 808–821.

Bishop, Christopher and Cazhaow Quazaz (1996). "Regression with input-dependent noise: A Bayesian treatment". In: *Advances in neural information processing systems* 9, pp. 347–353.

Carlin, Bradley P and Thomas A Louis (2000). *Bayes and empirical Bayes methods for data analysis*. Vol. 88. Chapman & Hall/CRC Boca Raton.

Cengel, Y and Transfer Mass Heat (2003). *A practical approach*. New York, NY, USA: McGraw-Hill.

Chaudhuri, Anirban et al. (2015). "Experimental flapping wing optimization and uncertainty quantification using limited samples". In: *Structural and Multidisciplinary Optimization* 51.4, pp. 957–970.

Clark Jr, Daniel L et al. (2016). "Engineering design exploration using locally optimized covariance Kriging". In: *AIAA Journal* 54.10, pp. 3160–3175.

Daniella, Dea (2015). “Experimental Optimization of 2-D Flapping Plate Kinematics Using Global Surrogate-Assisted Genetic Algorithm With Kriging”. MA thesis. Indonesia: Institut Teknologi Bandung.

Deb, Kalyanmoy et al. (2002). “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE transactions on evolutionary computation* 6.2, pp. 182–197.

Dwianto, Yohanes Bimo (2015). “Surrogate Assisted Genetic Algorithm for Expensive Fluid Flow Optimization”. MA thesis. Indonesia: Institut Teknologi Bandung.

Fathurrohim, Luqman (2016). “Maximizing the Thrust Performance of Flexible Caudal Fin Panels Using Global Surrogate Assisted Genetic Algorithm with Kriging”. MA thesis. Indonesia: Institut Teknologi Bandung.

Faza, Ghifari Adam (2018). “Multi-Objective Kriging-Based Optimization for Modern Wind Turbine Design Optimization”. MA thesis. Indonesia: Institut Teknologi Bandung.

— (2020). “Reduced Subspace Gaussian Process for Fast Probabilistic Analyses”. MA thesis. Indonesia: Institut Teknologi Bandung.

Fluck, Manuel and Curran Crawford (2017). “An engineering model for 3-D turbulent wind inflow based on a limited set of random variables”. In.

Forrester, Alexander IJ, Andy J Keane, and Neil W Bressloff (2006). “Design and analysis of" Noisy" computer experiments”. In: *AIAA journal* 44.10, pp. 2331–2339.

Gelman, Andrew et al. (2013). *Bayesian data analysis*. CRC press.

Gramacy, R.B. (2020). *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press. ISBN: 9781000766523. URL: `https://books.google.co.id/books?id=F9LVDwAAQBAJ`.

Hansen, Nikolaus, Sibylle D Müller, and Petros Koumoutsakos (2003). “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)”. In: *Evolutionary computation* 11.1, pp. 1–18.

Herbei, Radu and L. Mark Berliner (2014). “Estimating Ocean Circulation: An MCMC Approach With Approximated Likelihoods via the Bernoulli Factory”. In: *Journal of the American Statistical Association* 109.507, pp. 944–954. DOI: `10.1080/01621459.2014.914439`. eprint: `https://doi.org/10.1080/`

01621459.2014.914439. URL: https://doi.org/10.1080/01621459.2014.914439.

Hogg, Robert V, Elliot A Tanis, and Dale L Zimmerman (2010). *Probability and statistical inference*. Pearson/Prentice Hall Upper Saddle River, NJ, USA:

Huang, Deng et al. (2006). "Sequential kriging optimization using multiple-fidelity evaluations". In: *Structural and Multidisciplinary Optimization* 32.5, pp. 369–382.

Iqbal, Muhammad (2018). "Experimental Thrust Optimization of A Variable-Stiffness Fin Panel Using Global Surrogate-Assisted Genetic Algorithm With Kriging". MA thesis. Indonesia: Institut Teknologi Bandung.

Izzaturahman, Faiz (2020). "On A Support Vector Machine Approach to Surrogate Modelling: Predicting the Aeroelastic Flutter Boundary Within The Transonic Mach Regime". MA thesis. Indonesia: Institut Teknologi Bandung.

Jaluria, Y. (2002). *Computational Heat Transfer*. Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Taylor & Francis. ISBN: 9781560324775. URL: https://books.google.co.id/books?id=8Pv-D9U0yNoC.

Jim, Timothy Man Shui et al. (2020). "Bayesian Methods for Multi-Objective Optimization of a Supersonic Wing Planform". In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*. GECCO '20. Cancún, Mexico: Association for Computing Machinery, 1641–1643. ISBN: 9781450371278. DOI: 10.1145/3377929.3398122. URL: https://doi.org/10.1145/3377929.3398122.

Kanazaki, Masahiro et al. (2006). "Design exploration of high-lift airfoil using Kriging model and data mining technique". In.

Kersting, Kristian et al. (Jan. 2007). "Most-likely heteroscedastic Gaussian process regression". In: vol. 227, pp. 393–400.

Laurenceau, Julien and P Sagaut (2008). "Building efficient response surfaces of aerodynamic functions with kriging and cokriging". In: *AIAA journal* 46.2, pp. 498–507.

Lázaro-Gredilla, Miguel and Michalis K. Titsias (2011). "Variational Heteroscedastic Gaussian Process Regression". In: *ICML*.

Luhur, Muhammad Ramzan et al. (2015). “Stochastic model for aerodynamic force dynamics on wind turbine blades in unsteady wind inflow”. In: *Journal of Computational and Nonlinear Dynamics* 10.4.

Martin, Nathan and Morteza Gharib (2018). “Experimental trajectory optimization of a flapping fin propulsor using an evolutionary strategy”. In: *Bioinspiration & biomimetics* 14.1, p. 016010.

McKeague, Ian et al. (July 2005). “Statistical inversion of South Atlantic circulation in an abyssal neutral density layer”. In: *Journal of Marine Research* 63, pp. 683–704. DOI: 10.1357/0022240054663240.

Menzer, Olaf et al. (2013). “Random errors in carbon and water vapor fluxes assessed with Gaussian Processes”. In: *Agricultural and forest meteorology* 178, pp. 161–172.

Mukhopadhyay, Tanmoy et al. (2017). “A critical assessment of Kriging model variants for high-fidelity uncertainty quantification in dynamics of composite shells”. In: *Archives of Computational Methods in Engineering* 24.3, pp. 495–518.

Olson, James A (2001). “The motion of fibres in turbulent flow, stochastic simulation of isotropic homogeneous turbulence”. In: *International Journal of Multiphase Flow* 27.12, pp. 2083–2103.

Palar, Pramudita S. and Koji Shimoyama (2017). “Multi-Fidelity Uncertainty Analysis in CFD Using Hierarchical Kriging”. In: *35th AIAA Applied Aerodynamics Conference*. DOI: 10.2514/6.2017-3261. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2017-3261. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2017-3261.

Palar, Pramudita S. et al. (n.d.). “On the Impact of Covariance Functions in Multi-Objective Bayesian Optimization for Engineering Design”. In: *AIAA Scitech 2020 Forum*. DOI: 10.2514/6.2020-1867. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2020-1867. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2020-1867.

Palar, Pramudita Satria, Koji Shimoyama, and Lavi Rizki Zuhal (2020). “Uncertainty Quantification Methods for Evolutionary Optimization under Uncertainty”. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*. GECCO ’20. Cancún, Mexico: Association for Computing Machinery, 1614–1622. ISBN: 9781450371278. DOI: 10.1145/3377929.3398114. URL: https://doi.org/10.1145/3377929.3398114.

Palar, Pramudita Satria et al. (2016). "Framework for Robust Optimization Combining Surrogate Model, Memetic Algorithm, and Uncertainty Quantification". In: *Advances in Swarm Intelligence*. Ed. by Ying Tan, Yuhui Shi, and Ben Niu. Cham: Springer International Publishing, pp. 48–55. ISBN: 978-3-319-41000-5.

Palar, Pramudita Satria et al. (2019). "Benchmarking Constrained Surrogate-Based Optimization on Low Speed Airfoil Design Problems". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO '19. Prague, Czech Republic: Association for Computing Machinery, 1990–1998. ISBN: 9781450367486. DOI: 10.1145/3319619.3326829. URL: https://doi.org/10.1145/3319619.3326829.

Picheny, Victor, Tobias Wagner, and David Ginsbourger (2013). "A benchmark of kriging-based infill criteria for noisy optimization". In: *Structural and Multidisciplinary Optimization* 48.3, pp. 607–626.

Quadrianto, Novi et al. (2009). "Kernel conditional quantile estimation via reduction revisited". In: *2009 Ninth IEEE International Conference on Data Mining*. IEEE, pp. 938–943.

Rahmad, Yodefia (2019). "Balanced Inverse-Direct Design of Airfoil via Multi-Objective Optimization". MA thesis. Indonesia: Institut Teknologi Bandung.

Rai, Man Mohan and Nateri K. Madavan (2000). "Aerodynamic Design Using Neural Networks". In: *AIAA Journal* 38.1, pp. 173–182. DOI: 10.2514/2.938. eprint: https://doi.org/10.2514/2.938. URL: https://doi.org/10.2514/2.938.

Rasmussen, Carl Edward and Christopher K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. ISBN: 026218253X.

Rawlings, John O, Sastry G Pantula, and David A Dickey (2001). *Applied regression analysis: a research tool*. Springer Science & Business Media.

Raymer, Daniel P (2006). "Aircraft design: a conceptual approach and rds-student". In: *Software for Aircraft Design, Sizing, and Performance Set (AIAA Education), AIAA (American Institute of Aeronautics & Ast*.

Robani, Muhammad Daffa (2019). “Kriging-Based Infill Criteria for Expensive Optimization in Aerospace Problems”. MA thesis. Indonesia: Institut Teknologi Bandung.

Ruppert, David (1997). “Empirical-bias bandwidths for local polynomial non-parametric regression and density estimation”. In: *Journal of the American Statistical Association* 92.439, pp. 1049–1062.

Ruppert, David et al. (1997). “Local polynomial variance-function estimation”. In: *Technometrics* 39.3, pp. 262–273.

Sacks, Jerome et al. (1989). “Design and analysis of computer experiments”. In: *Statistical science*, pp. 409–423.

Sigrist, M. W., S. Bernegger, and P. L. Meyer (1989). “Atmospheric and Exhaust Air Monitoring by Laser Photoacoustic Spectroscopy”. In: *Photoacoustic, Photothermal and Photochemical Processes in Gases*. Ed. by Peter Hess. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 173–211. ISBN: 978-3-642-83851-4.

Silverman, Bernhard W (1985). “Some aspects of the spline smoothing approach to non-parametric regression curve fitting”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 47.1, pp. 1–21.

Sun, Guangyong et al. (2014). “Robust optimization of foam-filled thin-walled structure based on sequential Kriging metamodel”. In: *Structural and Multidisciplinary Optimization* 49.6, pp. 897–913.

Tan, Matthias HY (2016). “Monotonic quantile regression with Bernstein polynomials for stochastic simulation”. In: *Technometrics* 58.2, pp. 180–190.

Villani, Cédric (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

Wang, Chunyi (2014). *Gaussian process regression with heteroscedastic residuals and fast MCMC methods*. University of Toronto (Canada).

Wang, Yong Zhi et al. (Apr. 2015). “Composite Wind Turbine Blade Aerodynamic and Structural Integrated Design Optimization Based on RBF Meta-Model”. In: *Advanced Composites for Marine Engineering*. Vol. 813. Materials Science Forum. Trans Tech Publications Ltd, pp. 10–18. DOI: 10.4028/www.scientific.net/MSF.813.10.

Weinmeister, Justin, Xinfeng Gao, and Sourajeet Roy (2019). “Analysis of a Polynomial Chaos-Kriging Metamodel for Uncertainty Quantification in Aerodynamics”. In: *AIAA Journal* 57.6, pp. 2280–2296.

Winkelbauer, Andreas (2012). “Moments and absolute moments of the normal distribution”. In: *arXiv preprint arXiv:1209.4340*.

Zakaria, Kemas (2017a). “Mesh Deformation based on Radial Basis Function Interpolation for Aerodynamics Shape Optimization”. MA thesis. Indonesia: Institut Teknologi Bandung.

— (2017b). “Mesh Deformation based on Radial Basis Function Interpolation for Aerodynamics Shape Optimization”. MA thesis. Indonesia: Institut Teknologi Bandung.

— (2019). “Gradient-Enhanced Universal Kriging with Polynomial Chaos Expansion for Design Exploration”. MA thesis. Indonesia: Institut Teknologi Bandung.

Zhang, Q. and Y. Ni (2020). “Improved Most Likely Heteroscedastic Gaussian Process Regression via Bayesian Residual Moment Estimator”. In: *IEEE Transactions on Signal Processing* 68, pp. 3450–3460.

Zhang, Qiu-Hu and Yi-Qing Ni (2020). “Improved Most Likely Heteroscedastic Gaussian Process Regression via Bayesian Residual Moment Estimator”. In: *IEEE Transactions on Signal Processing* 68, pp. 3450–3460.

Zhang, Yiming et al. (2018). “Multifidelity surrogate based on single linear regression”. In: *AIAA Journal* 56.12, pp. 4944–4952.

Zhu, Shuang et al. (2020). “An improved long short-term memory network for streamflow forecasting in the upper Yangtze River”. In: *Stochastic Environmental Research and Risk Assessment*, pp. 1–17.

Zhu, X. and B. Sudret (2020). *Emulation of stochastic simulators using generalized lambda models*. arXiv: `2007.00996 [stat.CO]`.

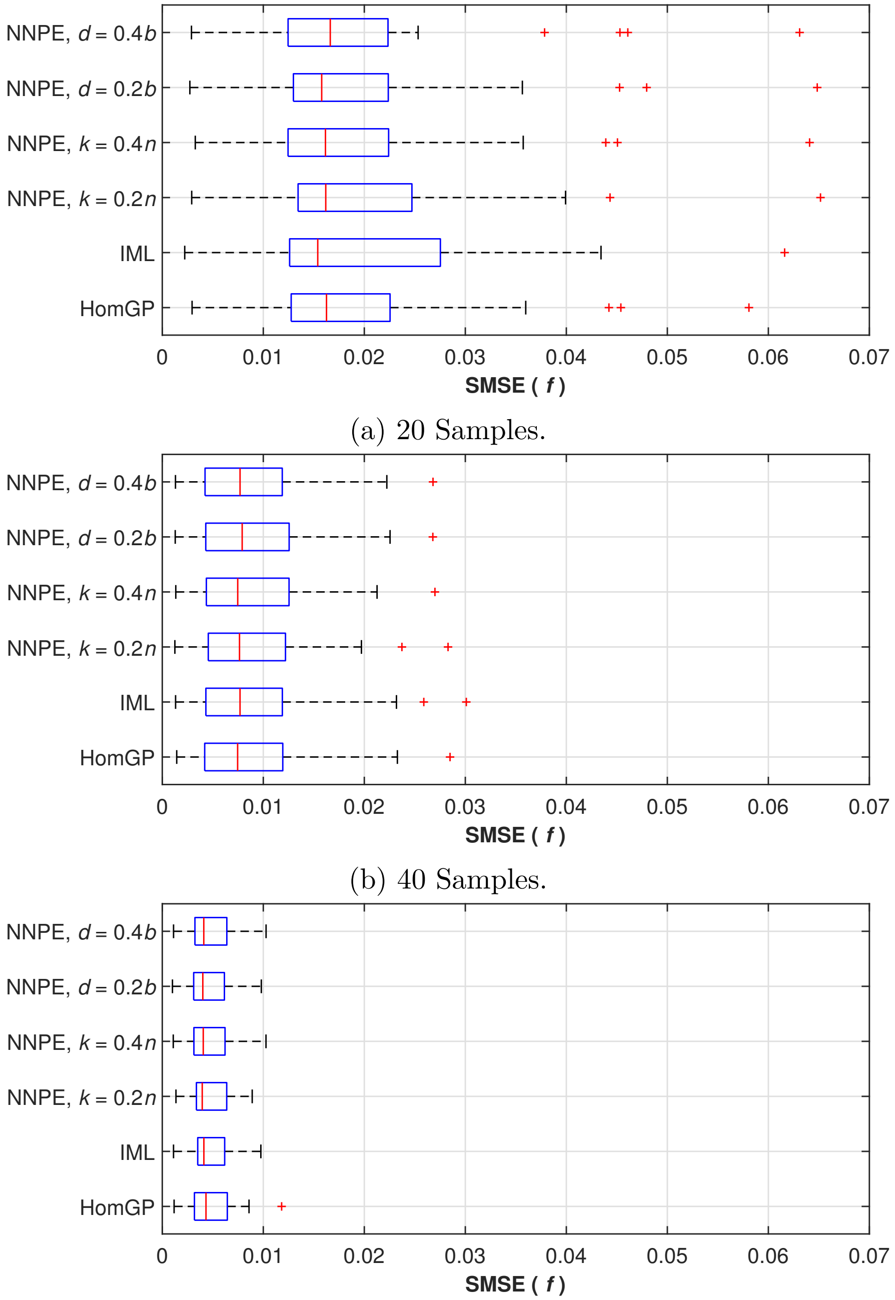
Zuhal, Lavi R. et al. (n.d.[a]). “Benchmarking Multi-Objective Bayesian Global Optimization Strategies for Aerodynamic Design”. In: *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. DOI: `10.2514/6.2018-0914`. eprint: `https://arc.aiaa.org/doi/pdf/10.2514/6.2018-0914`. URL: `https://arc.aiaa.org/doi/abs/10.2514/6.2018-0914`.

Zuhal, Lavi R. et al. (n.d.[b]). “Gradient-Enhanced Universal Kriging with Polynomial Chaos as Trend Function”. In: *AIAA Scitech 2020 Forum*. DOI: `10.2514/6.2020-1865`. eprint: `https://arc.aiaa.org/doi/pdf/10.2514/6.2020-1865`. URL: `https://arc.aiaa.org/doi/abs/10.2514/6.2020-1865`.

Zuhal, Lavi R. et al. (n.d.[c]). “Multi-Objective Kriging-Based Optimization for High-Fidelity Wind Turbine Design”. In: *AIAA Scitech 2019 Forum*. DOI: 10.2514/6.2019-0539. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-0539. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2019-0539.

Zuhal, Lavi Rizki, Pramudita Satria Palar, and Koji Shimoyama (2019). “A comparative study of multi-objective expected improvement for aerodynamic design”. In: *Aerospace Science and Technology* 91, pp. 548 –560. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2019.05.044. URL: http://www.sciencedirect.com/science/article/pii/S1270963818327809.

Appendix

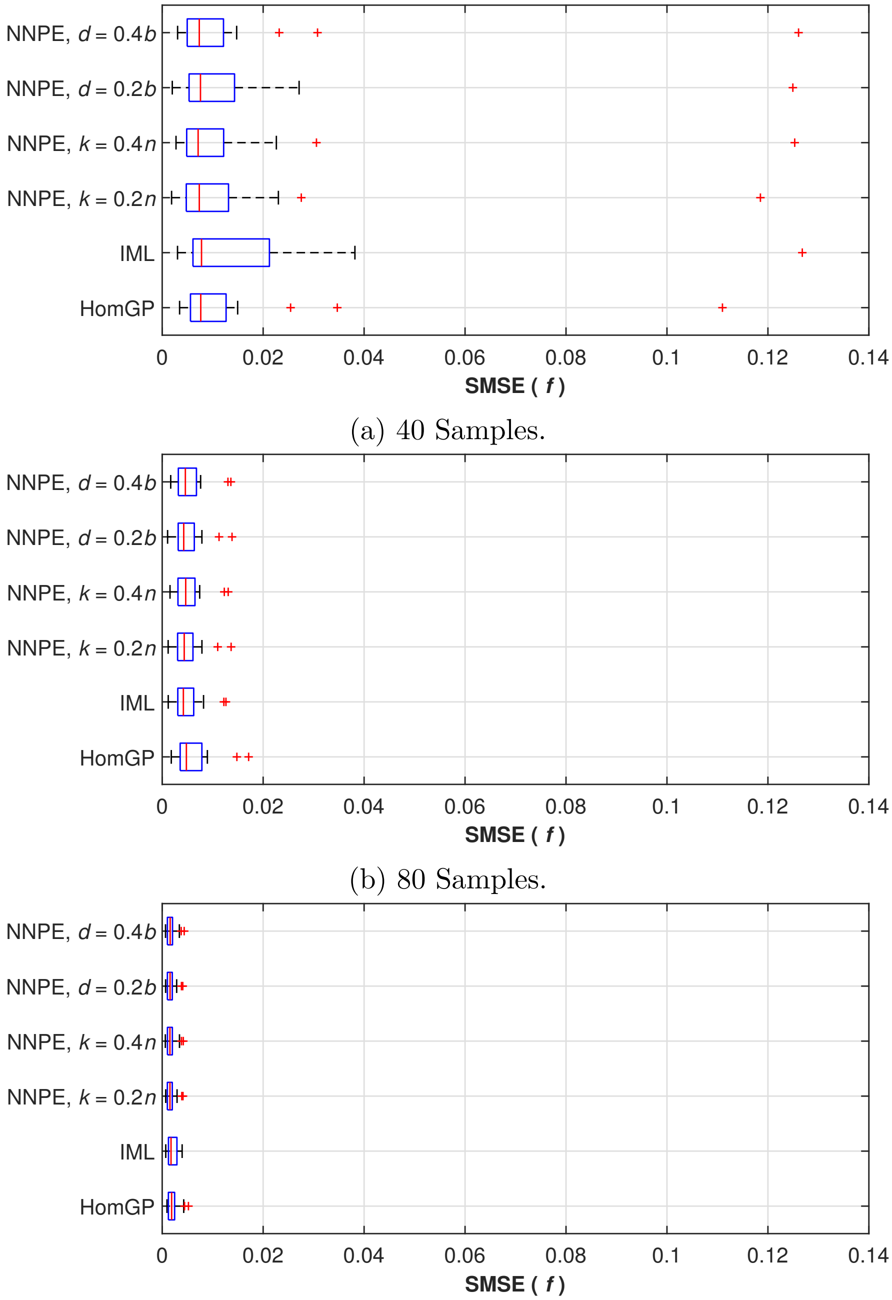


(a) 20 Samples.

(b) 40 Samples.

(c) 80 Samples.

Figure V.1: SMSE (f) results for one-dimensional mathematical function.

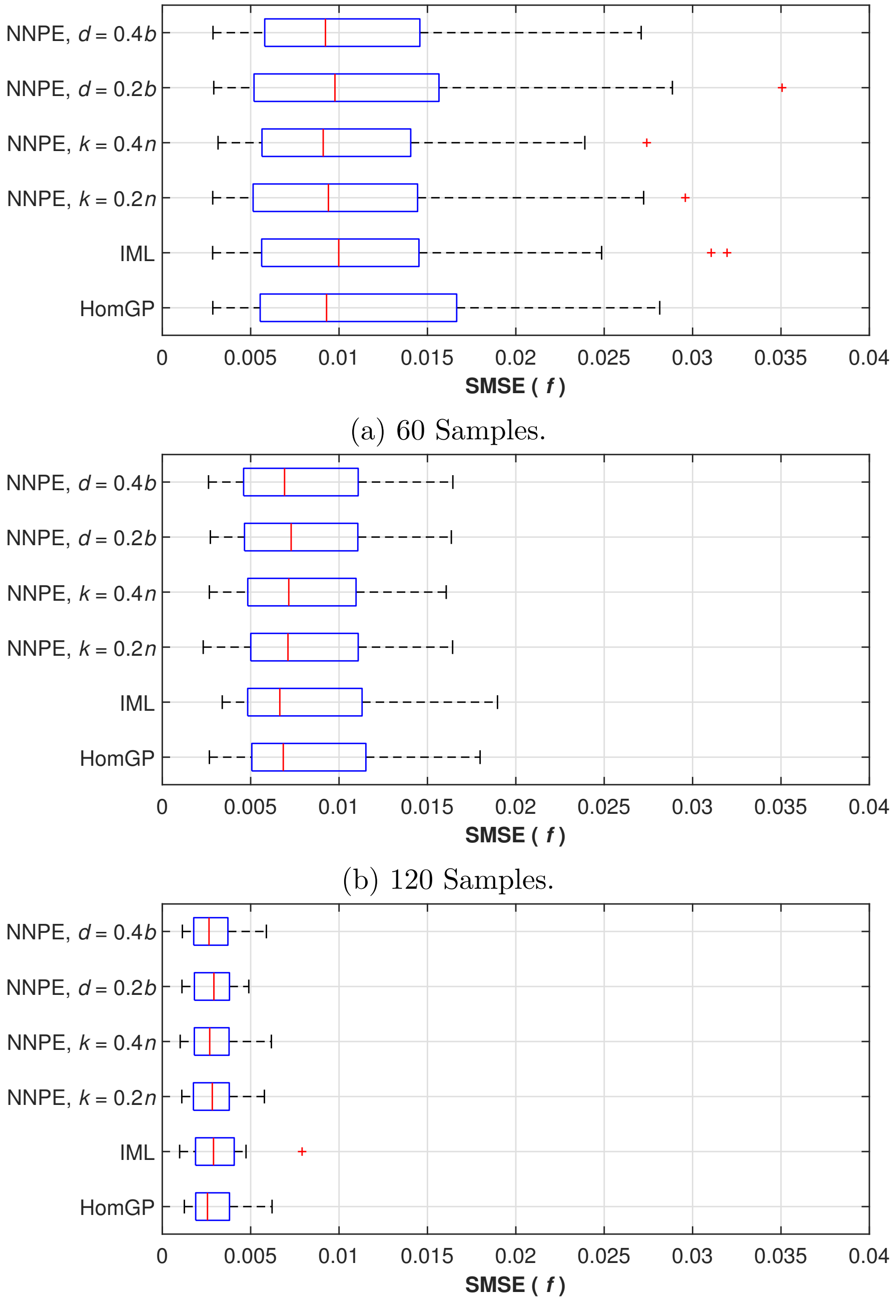


(a) 40 Samples.

(b) 80 Samples.

(c) 160 Samples.

Figure V.2: SMSE (f) results for two-dimensional mathematical function.

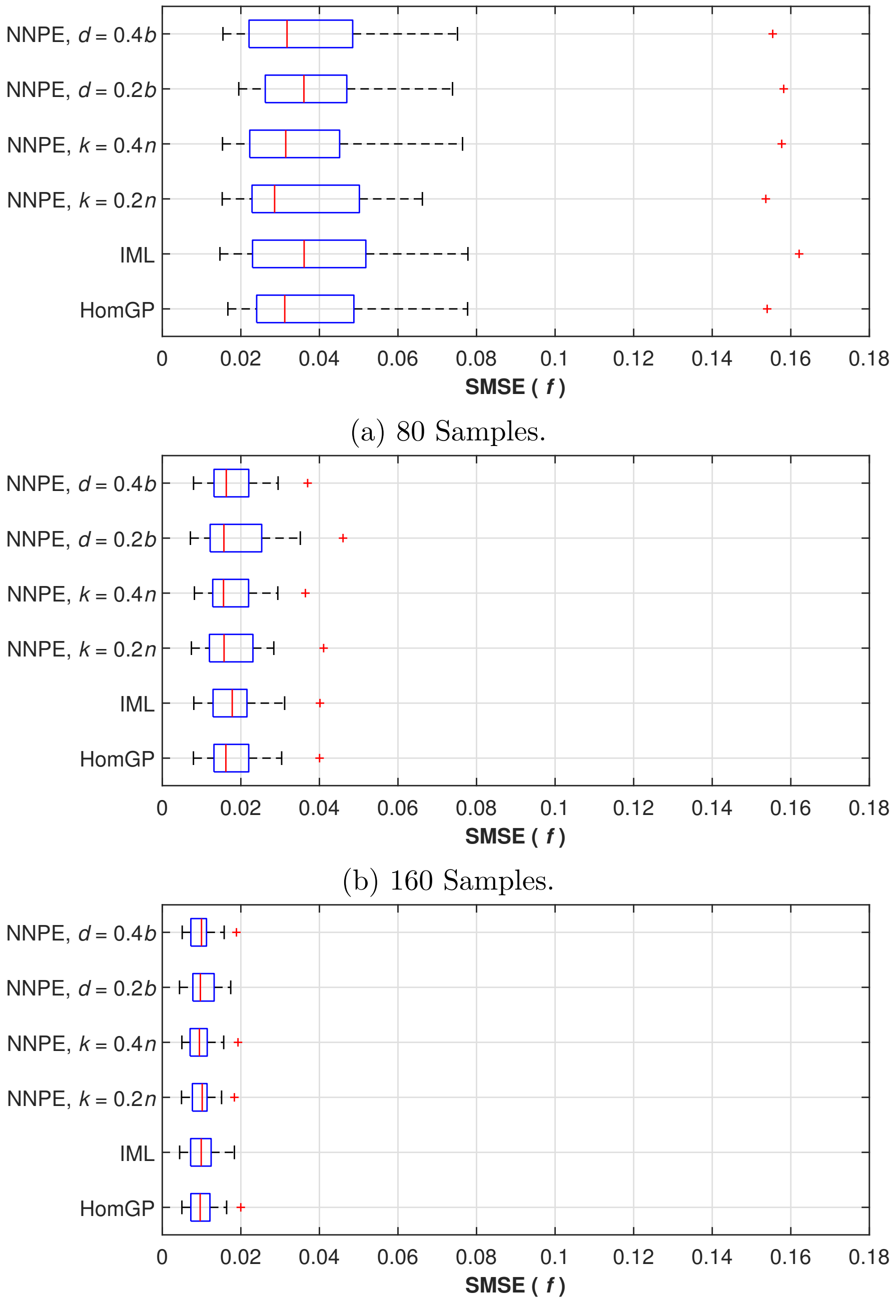


(a) 60 Samples.

(b) 120 Samples.

(c) 240 Samples.

Figure V.3: SMSE (f) results for three-dimensional operational envelope problem.

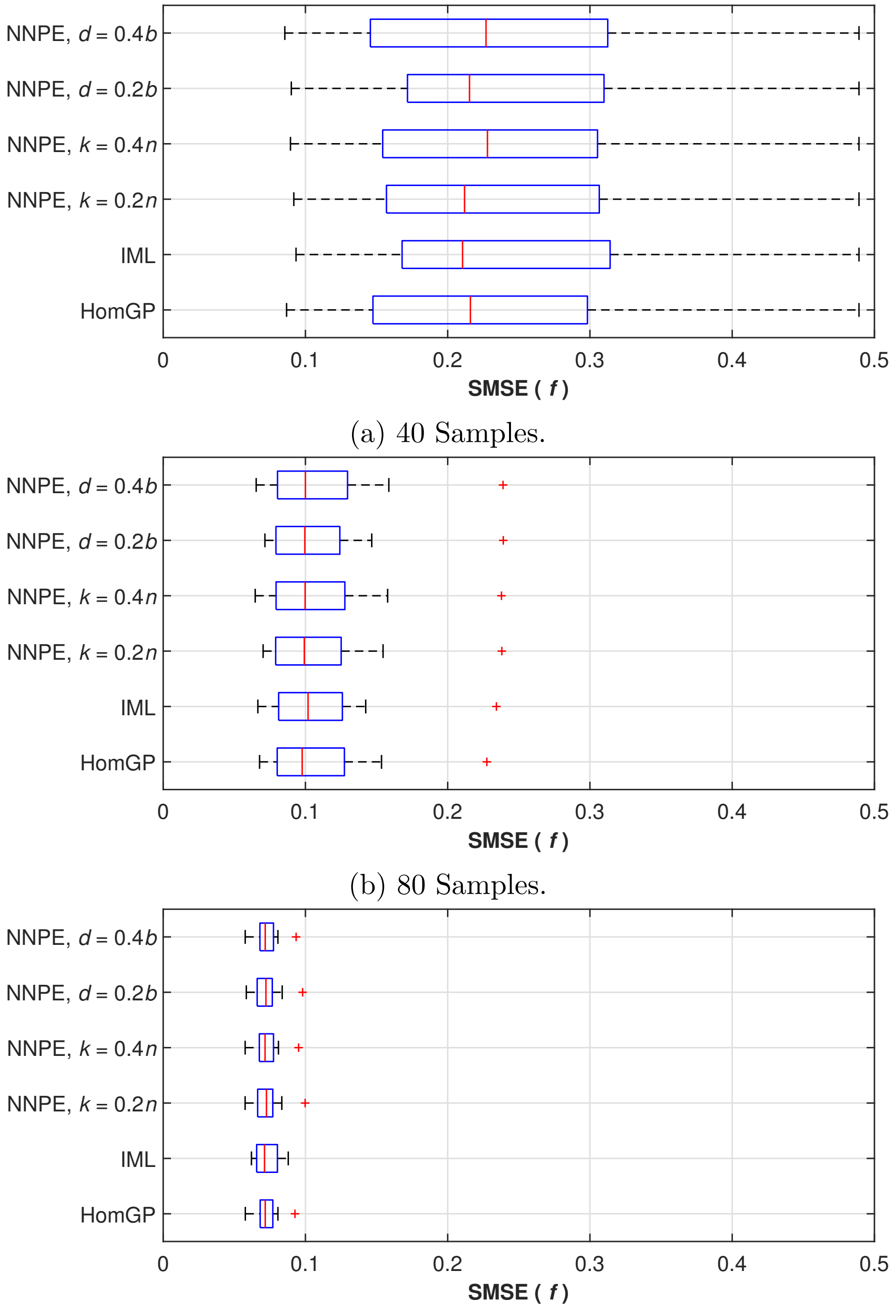


(a) 80 Samples.

(b) 160 Samples.

(c) 320 Samples.

Figure V.4: SMSE (f) results for four-dimensional temperature modeling problem.



(a) 40 Samples.

(b) 80 Samples.

(c) 160 Samples.

Figure V.5: SMSE (f) results results for the stochastic simulator problem.