Robust Gaussian Process Regression with a mixture of two Gaussian distributions as a noise model

by

Atefeh Daemi

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

Increasingly many complex processes from the different fields of biological systems, engineering or econometrics are often required to be controlled. Hence, in such cases, we deal with identification of underlying complex processes which is essential for control design, optimization, and process monitoring. However, developing models for complex processes purely based on first principles is a tedious task and sometimes infeasible. Data driven modelling which makes inference about the underlying process based on observations has been considered as a promising alternative in such scenarios. In data driven modelling, a mathematical model describing the relationship between observed measurements is obtained and thus identified model can be utilized to derive equations for prediction of unobserved values. In this thesis, Gaussian process (GP) as a non-parametric model which is a powerful approach to modelling of complex datasets is investigated from a Bayesian point of view.

One of the most important applications of Gaussian process models is in regression problems, wherein the output noise is commonly assumed to follow a normal distribution. However, in many practical problems, this assumption is not always realistic. Thus, we propose robust Bayesian methods to reduce the difference between the underlying process and the model, arising from outliers or other disturbances. We propose a mixture of two Gaussian distributions as a non-Gaussian likelihood for the noise model to capture both regular noise and irregular noise, so-called outliers, thereby making the regression model to be robust to the occurrence of outliers. We present an Expectation Maximization (EM) algorithm-based approach to making approximate inference possible for learning the proposed robust GP regression model. The proposed method is compared with other robust regression GP existing in the literature from a predictive performance perspective.

In this thesis, we also explore the problem of a new robust GP regression in

which the input presented to the model is noisy. To address this problem, we assume that the input noise is of an independently and identically distributed (i.i.d.) Gaussian noise and the output noise model is assumed to be distributed according to a mixture of two Gaussian distributions to capture both regular and irregular noises. We utilize the Expectation Maximization (EM) based algorithm that involves the errors-in-variables (EIV) to approximate the predictive distribution with a Gaussian process whose kernel function relies on both the input noise and the output noise hyper-parameters. Further, the improved performance of our proposed method is demonstrated by several illustrative examples.

The proposed robust GP with a Gaussian mixture noise model is also utilized for modelling nonlinear dynamic systems. In time series models based on the robust GP, we assume that the underlying process maps past observations and external inputs to the current observation, wherein the proposed robust GP with noisy input is employed for the multiple steps ahead prediction. It means that the whole predictive distribution of the output at any time step is fed back into the model for the next time-step which is considered as a noisy input to the model. Thus, the proposed model for nonlinear functions with input and output noise is used to learn true dynamics of the system which has been corrupted by outliers and to predict the output for multiple steps ahead in time. The effectiveness of the proposed approach is illustrated on both synthetic data and simulated Mackey-Glass chaotic time-series.

Acknowledgements

For the last two years, I have been given the privilege of studying at one of the best universities in Canada and had the opportunity of interacting with some of the best researchers in Control Engineering. I would like to express my gratitude to a number of kind people around me whom I have interacted with during my research, Since I could not have completed this manuscript without their helps and support.

First and foremost, I would like to thank my supervisor, professor Biao Huang for his patience, unconditional support and dexterity to guide me through every step of my graduate studies. I am particularly grateful for the trust he put in me to work on statistical machine learning. While I was first not familiar with this field his constructive comments and insightful feedback guided me whenever I faced difficulties during my research. I am also thankful to him for helping me to understand the concepts and makes me able to develop my own idea. My special thanks will also go to Dr.Hariprasad Kodamana, Dr.Yousef Alipouri, and Dr.Mohammad Rashed for giving valuable comments and support throughout my research journey. I greatly appreciate their help in reviewing my thesis.

As a part of the Computer Process Control (CPC) group, I would like to thank other members of the CPC group from whom I gained a lot of help and advice. I would like to thank Ruomu Tan for sharing her knowledge and expertise with me. I appreciated her help for guiding me in every aspect of academic life. I'm also thankful to all my colleagues for providing such a friendly and stimulating environment as especially: Rishik Ranjan, Yanjun Ma, Yaojie Lu, Shabnam Sedghi, Anahita Sadeghian, Dr. Fadi Ibrahim, Chaoqun Li, Mengqi Fang, Lei Fan, and other present and previous members of the CPC group.

I also want to express my gratitude to my true friends; Farzad, Parastoo, Amin, Amir and Fateme who never got tired of listening to my worries and complaints and

made everything better with their suggestions. I also acknowledge the help of my dear friend, Hossein for reviewing my thesis and constantly motivating me through this journey. I am thankful to my friend, Babak, for being my friend for almost 6 years, helping me find happiness in being the person that I really am.

Last but not least, I owe my immense gratitude to my parents for their boundless love and support. I would like to extend my gratitude to my brother, Aliakbar, and my twin sister, Atieh, for being my role models and the most influential characters in my life. I immensely thank them for their inspiration and giving me confidence in my abilities.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In the field of control theory, the main interest is to develop a model for the control of engineering processes. Control models can be characterized by different modelling techniques such as (i) mathematical models based on first principles (so-called knowledge driven modelling) whereby a control law is derived, and (ii) empirical models which make inference about the unknown underlying process based on time series analysis not from physics of the process (so-called Data driven modelling). We are often dealing with complex systems in chemical engineering which cannot be modelled with a simple mathematical model. Hence, data driven modelling from the field of machine learning is brought up in control community to enable us to control the complex systems, which is the motivation for this thesis.

Data driven modelling is usually based on parametric or non-parametric mathematical models. In this thesis, we focus on Gaussian process (GP) as a non-parametric model. In the traditional GP, the noise distribution model is of a Gaussian distribution. However, many processes may be affected by outlying observations or any other disturbances, resulting in a significant discrepancy between the true process and the model. The first objective of this thesis which is included in Chapter 2, is to increase the accuracy of the data driven modelling using a robust GP wherein we assume a mixture of two Gaussians noise model to capture both regular noise and outlying observations.

The motivation for the work to be presented in Chapter 3 is the fact that, in practice, a fault or malfunction in sensors, may lead to deficiency in the input mea-

surements. The identification model with noisy input has been studied in many statistic literature under the name of Error in variable (EIV) (Fuller [2009] ;Cheng et al. [1999]). The second objective of this thesis is to address this problem by the identification of a robust GP regression model with noisy input.

For control application, dynamic systems are required to be modelled by a mathematical relationship between past and current observations. Thus, we employ our aforementioned works for dynamic modelling in Chapter 4 to learn dynamics of the system which is corrupted with outliers.

1.2 Thesis Contributions

This thesis contributes mainly to the identification of both static and dynamic system using robust Gaussian process (GP) with a mixture of Gaussian likelihoods. Detailed contributions of this thesis can be summarized as follows:

1. Proposed a robust GP model with a mixture of two Gaussians noise model. This non-Gaussian likelihood as a noise model capture both regular noise and outliers. Further, An EM algorithm based approach as an alternative to maximum likelihood estimation (MLE) approach is proposed to learn the hyper-parameters of the GP and mixture noise by constructing a lower bound to MLE. Thus, the predictive distribution for unobserved measurements is found based on the parameters estimated by EM.

2. Considered a new robust GP model in which both input and output are corrupted by the two types of noises. We used the normal distribution to describe the input noise and a mixture of two Gaussian distributions to represent the output noise for considering both outlying observations and regular noises in output. Then, a novel learning scheme based on EM algorithm and error in variables modelling is proposed.

3. Utilized the proposed robust GP regression with a mixture of two Gaussians noise model for identification of nonlinear dynamic systems. We considered NARX model wherein the proposed robust GP is placed as a prior on the nonlinear function mapping regressors including past outputs and external inputs

to the current output. Using this robust identification method, we can deal with a time series which is corrupted by outliers and learn the true dynamics of this system.

1.3 Thesis Outline

The layout of the thesis is organized as follows:

In Chapter 2, we propose a robust GP regression model with a mixture of two Gaussians noise model. An EM algorithm based approach is employed to learn optimal point estimation of the hyper-parameters of the proposed robust GP. Further, the first and second moment of the predictive distribution for unobserved values are derived.

Chapter 3 introduces the task of modelling with a robust Gaussian process with a mixture of two Gaussians noise model in the presence of a noisy input. This problem is solved by taking an EM-based approach that consists of approximating the prior on the underlying process with a new Gaussian process.

In Chapter 4, the proposed robust GP with a mixture of two Gaussians noise model is utilized for Dynamic system modelling within NARX structure as a robust system identification which targets the data with outlying observation.

Chapter 5 concludes the main results of the thesis and suggests some opportunities for future work.

This thesis adopts the paper format. Each chapter will have its own introduction, literature review and conclusion sections.

Chapter 2

Robust Gaussian Process Regression with a Gaussian Mixture Likelihood using Em Algorithm [1]

Gaussian Process (GP), as a probabilistic nonlinear multi-variable regression model, has been widely used in non-parametric Bayesian framework for the data based modelling of complex processes. The output noise in Standard GP regression is assumed to follow a Gaussian distribution. In this setting, the point estimation of the model parameters can be obtained analytically using the maximum likelihood (ML) approach in a straight forward fashion. However, in practical scenarios, processes may have been corrupted by the outliers and other disturbances or have multiple modes of operation, resulting a non-Gaussian data likelihood. In this work, to model such scenarios, we propose to employ a mixture of two Gaussian distributions as the noise model to capture both regular noise and irregular noise, thereby enhancing the robustness of the regression model. Further, we present an Expectation Maximization (EM) algorithm-based approach to obtain the optimal parameters set of the proposed GP regression model. The predictive distribution can then be found according to the estimated hyper-parameters from the EM algorithm. The efficacy and practicality of the proposed method are illustrated with two sets of synthetic data, a simulated example, as well as an industrial data set.

[1] Submitted as A. Daemi, , K. Hariprasad, B. Huang,"Gaussian Process Modelling with Gaussian Mixture Likelihood". Journal of Process Control.

2.1 Introduction

Modelling of complex processes is essential for optimization, control, and process monitoring. However, developing first principles based models for complex chemical processes is a tedious task. Hence, data based models have been considered as a promising alternative in such scenarios. Recently, significant attention has been drawn into data-driven non parametric models as well. Some popular non-parametric regression models include Gaussian Process Regression (GPR), Support Vector Regression (SVR), Artificial Neural Network(ANN), among others. Non-parametric models can learn any functional form of models from the training data without any prior knowledge. They require only input-output sets of data alone for the modelling [Russell et al., 1995]. For instance, SVR, proposed by Vapnik [1995] as a regression method, constructs a hyperplane to maximize the separation between data points. Neural network models are typically structured in layers which include a number of interconnected nodes mimicking biological neural networks [Demuth et al., 2014].

Gaussian Process (GP), a non-parametric modelling paradigm, was initially introduced in the field of geo-statistics in the name "kriging" [Krige, 1951]. Kriging calculates the weights based on the inverse distance between the predicted values and the measured inputs as well as the spatial auto correlation of the measured inputs. The basic underlying assumption of GPR is that, a collection of any arbitrary function values can be modelled using multivariate Gaussian distribution [O'Hagan and Kingman, 1978]. It was shown by Neal [1995] that Bayesian neural networks with infinite hidden nodes in one layer is equivalent to GPs. Hence, it can be viewed as flexible and interpretable alternatives to neural networks. GP can also be derived from other models such as Bayesian kernel machines, and linear regression with basis functions [Williams, 1999]. Due to the computational difficulty of Bayesian analysis of neural networks [MacKay, 1992, Neal, 1993], GP was used by Williams and Rasmussen [1996] as a regression model to make the predictive Bayesian analysis straightforward. The Bayesian interpretation of GPs was further enriched and extended due to Neal [1997] and Gibbs [1998].

The ability to model complex data sets makes GPR promising in the area of data based process modelling. For instances, spectroscopic calibration [Chen et al., 2007],

development of soft sensors [Liu et al., 2015], state estimation of Lithium-ion batteries [He et al., 2015] and model predictive control [Murray-smith et al., 2004] have found it's application. A natural way to model such industrial data is by attributing Gaussian distribution to the noise. The fully Bayesian framework of GP is computationally tractable for the Gaussian noise model. However, in realistic scenarios, the industrial data seldom follows Gaussian distribution as it may contain outliers due to sensor malfunctions and process disturbances or due to data emanating from multiple operational modes of process. To deal with such scenarios, a possible work around is to employ non-Gaussian distributions for modelling the noise dynamics resulting in a more robust model [Box and Tiao, 1962]. Various approaches have been followed by different researchers for accommodating outliers while modelling the industrial process data. For instances, O'Hagan [1979] has discussed the distributions with thick tails and termed them "outlier- prone" as they reject outlying observations, and Jaynes [2003] proposed a two-model strategy containing a good and a bad sampling distribution to model regular and outlying observations. Further, in similar lines, use of Student's-t distribution, as a heavy tailed distribution to accommodate outliers, has been described by West [1984], a mixture of two Gaussian distributions is introduced by Box and Tiao [1968], and Laplace distribution was also used as a noise distribution in Rousseeuw and Leroy [2005].

In the context of GPR, Kuss [2006] investigated the possibility of Student's-t distribution for describing the noise model. Kuss [2006] applied variational inference, Expectation propagagation (EP) and Markov chain Monte Carlo(MCMC) methods for inference of the GPR model with Student's-t likelihood. This work is a further extension of approximate variational framework for regression [Tipping and Lawrence, 2005]. Moreover, Vanhatalo et al. [2009] used the Laplace's approximation for approximating log-marginal likelihood of the complete data, while Jylänki et al. [2011] proposed expectation propagation (EP) for the approximate inference of the GPR model with Student's t likelihood.

Recently, Ranjan et al. [2016] proposed an EM algorithm based approach for robust GPR identification using non-Gaussian noise distributions, namely, Student's-t and Laplace distribution.

In this work, we develop a GPR model with a mixture of two Gaussian distri-

butions as data likelihood. The considered model would capture scenarios like, data with outliers from a contaminated distribution as well data obtained from a process operating in multiple modes, which are not uncommon in chemical processes. Further, we propose to use EM algorithm to learn hyper-parameters of the proposed GPR model. The EM algorithm is a powerful approach for obtaining maximum likelihood estimates (MLE) and is useful when the observed data is incomplete or containing hidden or latent variables[Dempster et al., 1977]. Even though Kuss [2006] investigated the scenario for a mixture of two Gaussian noises model in GPR, the entire focus was on inference rather than determining the model's hyper-parameters. In this work we address this lacunae by deriving parameter estimates of GPR model for a mixture of Gaussian likelihood. To the best of the authors' knowledge, there exists no approach in literature to estimate the parameters of the GPR with a mixture of Gaussian likelihood. Finally, we also validate our results with two synthetic data sets, a simulated CSTR example and an industrial data set.

The rest of this chapter is organized as follows: Section 2.2 provides a revisit of GPR. The problem is described in Section 2.3 . In Section 2.4, an EM algorithm based approach is derived to estimate hyper-parameters of GPR. After learning the hyper-parameters, a procedure for prediction using test data is discussed in Section 2.5. Section 2.6 presents an algorithmic flowchart for the estimation of hyper-parameters. In Section 2.7, three validation studies are presented to verify the efficiency of the proposed GPR model. Summary of our findings and conclusions are provided in Section 2.8.

2.2 Revisit of GPR

The GPR modelling paradigm tries to find a distribution over a set of possible non-parametric functions for modelling a set of input and output data-sets. Traditionally, this relationship was characterized by various classes of parametric functions. Suppose we observe some inputs $\mathbf{x_i}$ and some outputs f_i, where $f_i = f(\mathbf{x_i})$ represents the unknown underlying mapping function.

Let $\mathbf{x_i} \in \mathbb{R}^d$ be the set of inputs for the i^{th} training sample. Then we define a new

Table 2.1: Gaussian Process Kernel And Mean Options

Kernel Function	Mathematical Expression, $k(\mathbf{x}, \mathbf{x}')$
Constant	σ^2
Linear	$\mathbf{x}.\mathbf{x}'$
Polynomial	$(\mathbf{x}.\mathbf{x}' + C)^p$
Squared Exponential or RBF	$\exp[-\frac{\lvert\mathbf{x}-\mathbf{x}'\rvert^2}{2l^2}]$
Matérn	$\frac{2^{1-\nu}}{\Gamma(\nu)}(\frac{\sqrt{2\nu}\lvert\mathbf{x}-\mathbf{x}'\rvert}{l})^\nu K_\nu(\frac{\sqrt{2\nu}\lvert\mathbf{x}-\mathbf{x}'\rvert}{l})$
Exponential	$\exp[-\frac{\lvert\mathbf{x}-\mathbf{x}'\rvert}{l}]$
γ-exponential	$\exp[-(\frac{\lvert\mathbf{x}-\mathbf{x}'\rvert}{l})^\gamma]$
Rational Quadratic	$(1 + \frac{\lvert\mathbf{x}-\mathbf{x}'\rvert}{2\alpha l^2})^{-\alpha}$
Mean Function	**Mathematical Expression, $m(\mathbf{x})$**
Zero	0
Constant	c
Linear	$\mathbf{x}.\boldsymbol{\alpha}^T$
Polynomial	$\sum_m \boldsymbol{\alpha}_{\boldsymbol{m}}^{\boldsymbol{T}}(\mathbf{x})^m$

variable $\mathbf{X}$ for the collection of n training samples, having d dimensions as follows:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1d} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2d} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nd} \end{bmatrix} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \vdots \\ \mathbf{x_i} \\ \vdots \\ \mathbf{x_n} \end{bmatrix}$$

A GP assumes a prior over the f_i at a finite set of points, $\mathbf{f} = (f(\mathbf{x_1}), f(\mathbf{x_2}), ..., f(\mathbf{x_n}))$ as

$$P(\mathbf{f}|\mathbf{X}, \theta_{GP}) \sim \mathcal{N}(\mathbf{m}(\mathbf{X}), \mathbf{K}(\mathbf{X}, \mathbf{X})), \tag{2.1}$$

to be jointly multivariate Gaussian [Ebden, 2015]. Any element of covariance matrix is given by $K_{ij} = K(\mathbf{x_i}, \mathbf{x_j})$, where k(.,.) is a positive definite kernel matrix. A kernel is a similarity function by which if $\mathbf{x_i}$ and $\mathbf{x_j}$ are considered similar, the output of the function at those points are expected to be similar. Most commonly used form of Kernel function is exponential. The table 2.1 provides some common kernel and mean functions. The interested readers, for details regarding various Kernel functions used in GP, are referred to Rasmussen and Williams [2006].

The complete specification of function $f(.)$ is provided by maximizing log marginal likelihood of the observation towards the parameters of mean and covariance function.

The joint distribution of unobserved output $\mathbf{f}_+$ at a test input set $\mathbf{X}_+$ based on characterization of the GP can be written as,

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_+ \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mathbf{m}(\mathbf{X}) \\ \mathbf{m}(\mathbf{X}_+) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) & \mathbf{K}(\mathbf{X},\mathbf{X}_+) \\ \mathbf{K}(\mathbf{X}_+,\mathbf{X}) & \mathbf{K}(\mathbf{X}_+,\mathbf{X}_+) \end{bmatrix}) \tag{2.2}$$

where covariance matrix and vector of means are constructed according to the specified parameters obtained from training. By the probability rules for conditioning Gaussian (refer to A.1), the posterior has the following form[Rasmussen, 1996]:

$$\begin{aligned} &p(\mathbf{f}_+|\mathbf{f},\mathbf{X},\mathbf{X}_+,\theta) \sim \mathcal{N}(\boldsymbol{n}_+,\mathbf{S}_+) \\ &\text{where} \qquad \boldsymbol{n}_+ = \mathbf{m}(\mathbf{X}_+) + \mathbf{K}(\mathbf{X}_+,\mathbf{X})\ \mathbf{K}(\mathbf{X},\mathbf{X})^{-1}(\mathbf{f}-\mathbf{m}(\mathbf{X})), \\ &\qquad\qquad \mathbf{S}_+ = \mathbf{K}(\mathbf{X}_+,\mathbf{X}_+) - \mathbf{K}(\mathbf{X}_+,\mathbf{X})\ \mathbf{K}(\mathbf{X},\mathbf{X})^{-1}\mathbf{K}(\mathbf{X},\mathbf{X}_+) \end{aligned} \tag{2.3}$$

As an example, the GP posterior of the function $x^{0.5}\cos x + \cos x^2$ is graphically presented in Fig.(2.1). As observed from Fig.(2.1), the advantage of GP modelling approach is that it also provides us with the uncertainty over prediction which is shown as the level of confidence in the predicted output[Pedregosa et al., 2011].

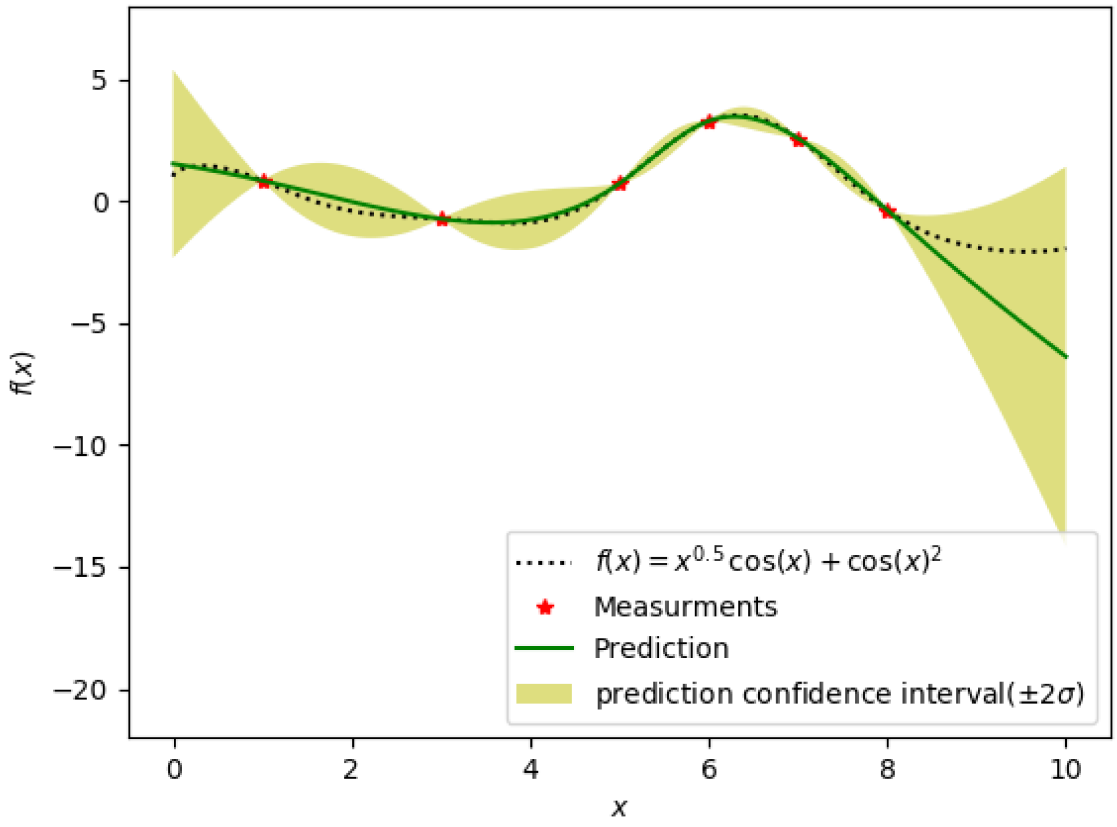

Figure 2.1: Mean prediction with the prediction interval by 2 standard deviation from the mean

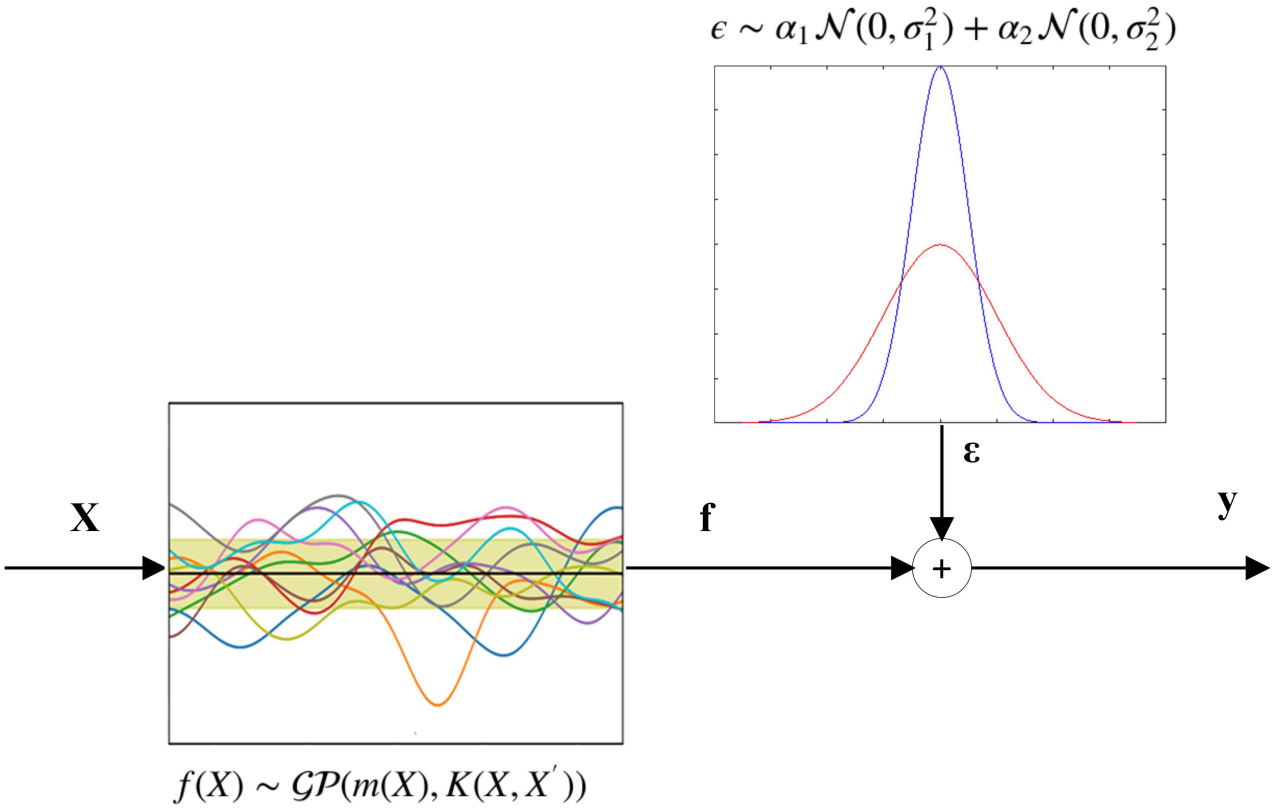

Figure 2.2: A schematic of GPR with a mixture of two Gaussian noises

2.3 Problem Statement

This section is allocated to describing the problem statement. Fig.(2.2) shows the graphical model for GPR with a mixture of two Gaussian noises. The GPR in Eq.(2.4) assumes that the existence of a latent function $f(\mathbf{x}, \theta)$ mapping the the deterministic input $\mathbf{x}$ to the noise free output,f, where θ are the set of underlying hyper-parameters. Also, y denotes the observed output which is disturbed by a noise. In this case the noise, ϵ, is assumed to be a mixture of two Gaussian distributions, as given below:

$$y = f(\mathbf{x}, \theta) + \epsilon \tag{2.4}$$

The actual output (f) from process has been corrupted with a mixture of Gaussian distributions noise (ϵ) resulting the noise corrupted output (y). For the sake of simplicity in the analytics we limit the number of Gaussian components to be two with zero mean and different variances σ_1^2 and σ_2^2 as below,

$$\epsilon \sim \begin{cases} \mathcal{N}\left(0, \sigma_1^2\right) & w.p.\ \alpha_1 \\ \mathcal{N}\left(0, \sigma_2^2\right) & w.p.\ \alpha_2 \end{cases} \tag{2.5}$$

In this setting, the observed data (C_{obs}) include $\mathbf{X}$ and $\mathbf{y} = [y_1\ y_2\ y_3\ \ldots\ y_n]^T$ where $y_i \in \mathbb{R}$ is a scalar output. The missing (or hidden) data (C_{mis}) include $\mathbf{f} = [f_1\ f_2\ f_3\ \ldots\ f_n]^T$ where f_i is the value of the latent function at input $\mathbf{x_i}$, and the mode identity of the noise at different samples, i.e., $\mathbf{I} = [I_1\ I_2\ I_3\ \ldots\ I_n]^T$ where I_i indicates that the noise of the i^{th} sample point is attributed to which noise component. For a Gaussian mixture distribution with two noise components as in our case, $I_i \in \{1, 2\}$.

We assume the distribution of errors is a mixture of a model with relatively high variance σ_1^2 accounting for the outlier distribution and a second model for the regular noise which has a small variance σ_2^2 compared to the variance of outliers. We use α_j to denote the probability of occurring of outliers or regular noises, resulting in $\sum_{j=1}^{2} \alpha_j = 1$. Eq.(2.6), presented below, shows the expression of Gaussian mixture noise model considered for this study:

$$P(y_i|f_i; \theta_n) = \sum_{j=1}^{2} \alpha_{I_i=j} \frac{1}{\sqrt{2\pi\sigma_{I_i=j}^2}} \exp[-\frac{(y_i - f_i)^2}{2\sigma_{I_i=j}^2}] \tag{2.6}$$

where $\theta_n = [\alpha_1, \sigma_1^2, \sigma_2^2]$ denotes the hyper-parameters involved in the Gaussian mixture noise model. The noise distribution for each sample $i = 1, ..., n$ given the noise mode identity $I_i = j$, can be written as,

$$P(y_i|f_i, I_i = j; \theta_n) = \frac{1}{\sqrt{2\pi\sigma_{I_i=j}^2}} \exp[-\frac{(y_i - f_i)^2}{2\sigma_{I_i=j}^2}] \tag{2.7}$$

where $\sigma_{I_i=j}^2$ is variance for the distribution with noise mode identity j. The equation can be further rewritten considering all the n samples, in the multivariate vector form, as below:

$$P(\mathbf{y}|\mathbf{f}, \mathbf{I}; \theta_n) = \frac{1}{\sqrt{(2\pi)^n |diag(\sigma_\mathbf{I})|}} \exp[-\frac{(\mathbf{y} - \mathbf{f})^T \Big(diag(\sigma_\mathbf{I})\Big)^{-2} (\mathbf{y} - \mathbf{f})}{2}] \tag{2.8}$$

In a GP model, the conditional distribution of $\mathbf{f}|\mathbf{X}$ is assumed to follow a multivariate Gaussian which is completely characterized by its first and second moments, i.e. mean function $\mathbf{m}(\mathbf{X})$ and the covariance function $\mathbf{K}(\mathbf{X}, \mathbf{X})$ presented in Eq.(2.1), whose covariance function, in this work, is assumed to be the squared exponential (SE) kernel,[Murphy, 2012]

$$k_{SE}(\mathbf{x_i}, \mathbf{x_j}) = \sigma_f^2 \ \exp(-\frac{1}{2}(\mathbf{x_i} - \mathbf{x_j})^T \mathbf{\Lambda^{-1}} (\mathbf{x_i} - \mathbf{x_j})) \tag{2.9}$$

where θ_{GP} denotes all the hyper-parameters of a GP that includes mean parameter, signal variance σ_f^2, and length-scale diagonal matrix, $\mathbf{\Lambda} = diag([{l_1}^2, {l_2}^2, .., {l_d}^2])$ where d denotes the input-space dimensions. The covariance function $k(\mathbf{X}, \mathbf{X})$ specifies the covariance between pairs of the outputs as a function of inputs. Then the prior distribution on $\mathbf{f}$ can be rewritten as:

$$P(\mathbf{f}|\mathbf{X}; \theta_{GP}) = \frac{1}{\sqrt{|2\pi\mathbf{K}(\mathbf{X}, \mathbf{X})|}} \exp[-\frac{(\mathbf{f} - m\mathbf{1})^T (\mathbf{K}(\mathbf{X}, \mathbf{X}))^{-1} (\mathbf{f} - m\mathbf{1})}{2}] \tag{2.10}$$

where m is a scalar value for the constant mean function and $\mathbf{K}(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n \times n}$ is a matrix and $\mathbf{1}$ represents a vector of ones with appropriate dimensions.

Let the complete set of the hyper-parameters be $\vartheta = [\theta_{GP}, \theta_n]$. If we employ ML estimation to estimate the parameter set ϑ, we will have the following results:

$$\hat{\vartheta} = \arg\max_{\vartheta} P(\mathbf{y}|\mathbf{X}, \vartheta) \tag{2.11}$$

and by marginalization and Chain rule of probability,

$$= \arg\max_{\vartheta} \sum_{I} \int_{f} P(\mathbf{y}, \mathbf{f}, \mathbf{I}|\mathbf{X}, \vartheta)\ d\mathbf{f} \tag{2.12}$$

$$= \arg\max_{\vartheta} \sum_{I} \int_{f} P(\mathbf{y}|\mathbf{f}, \mathbf{I}, \mathbf{X}, \vartheta) P(\mathbf{f}|\mathbf{I}, \mathbf{X}, \vartheta) P(\mathbf{I}|\mathbf{X}, \vartheta)\ d\mathbf{f} \tag{2.13}$$

$$= \arg\max_{\vartheta} \sum_{I} \int_{f} P(\mathbf{y}|\mathbf{f}, \mathbf{I}, \vartheta) P(\mathbf{f}|\mathbf{X}, \vartheta) P(\mathbf{I}|\vartheta)\ d\mathbf{f} \tag{2.14}$$

In Eq.(3.5), the likelihood of $\mathbf{I}|\vartheta$ follows binomial distribution for the mixture of two Gaussian components. As there are two possibilities for each observation, for n samples, a total of 2^n number of combinations need to be enumerated, leading to combinatorial problems. Therefore, the evidence likelihood is analytically intractable, and the parameters cannot be directly estimated through the ML approach. Hence, as an alternative, Expectation-Maximization (EM) algorithm is employed in this work to iteratively obtain the ML parameter estimates. After obtaining all the hyper-parameters, the posterior predictive distribution can be determined. These steps are presented in detail in the next section.

2.4 Parameter Estimation using the EM algorithm

The EM algorithm consists of the following iterative steps, which are repeated till convergence (Lu and Huang [2014];Guo et al. [2017]) to obtain approximate ML estimates:

- **E-Step:** In this step, the expectation of the logarithm of the likelihood probability of all hidden and observed data with respect to conditional distribution of hidden data given observed data and current estimate of the hyper-parameters, called Q-function, will be derived:

$$Q(\vartheta; \vartheta^{(t)}) = E_{C_{mis}|C_{obs},\vartheta^{(t)}}[\log(P(C_{obs}, C_{mis}|\vartheta))] \tag{2.15}$$

- **M-Step:** In this step, the Q-function is maximized to obtain the parameter estimates:

$$\vartheta^{(t+1)} = \arg\max_{\vartheta}\ Q(\vartheta; \vartheta^{(t)}) \tag{2.16}$$

where $\vartheta^{(t)}$ refers to the parameters estimated in t-th iteration. For detail information regarding EM algorithm, readers are referred to(Borman [2004]).

2.4.1 E-Step

The Q function of our problem can be formulated as,

$$Q(\vartheta;\vartheta^{(t)}) = E_{\mathbf{f},\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}[\log(P(\mathbf{y},\mathbf{f},\mathbf{X},\mathbf{I}|\vartheta))] \tag{2.17}$$

By factorizing the joint probability distribution of $P(\mathbf{y},\mathbf{f},\mathbf{X},\mathbf{I}|\vartheta)$ according to chain rule,

$$\begin{aligned} P(\mathbf{y},\mathbf{f},\mathbf{X},\mathbf{I}|\vartheta) &= P(\mathbf{y}|\mathbf{f},\mathbf{X},\mathbf{I},\vartheta)P(\mathbf{f}|\mathbf{X},\mathbf{I},\vartheta)P(\mathbf{I}|\mathbf{X},\vartheta)P(\mathbf{X}|\vartheta) \\ &= P(\mathbf{y}|\mathbf{f},\mathbf{I},\theta_n)P(\mathbf{f}|\mathbf{X},\theta_{GP})P(\mathbf{I}|\theta_n) \end{aligned} \tag{2.18}$$

Substituting Eq.(2.18) into Eq.(2.17) and using the properties of the log operator, the Q function becomes,

$$Q(\vartheta;\vartheta^{(t)}) = E_{\mathbf{f},\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\{\log(P(\mathbf{y}|\mathbf{f},\mathbf{I},\theta_n)) + \log(P(\mathbf{f}|\mathbf{X},\theta_{GP})) + \log(P(\mathbf{I}|\theta_n))\} \tag{2.19}$$

The expectation calculation (2.19) can be broken down into two parts. First, the expected value of each term with respect to the posterior $\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}$ is calculated; then the expectation over $\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}$ is performed, as below,

$$\begin{aligned} Q(\vartheta;\vartheta^{(t)}) = E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{ & E_{\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}}\{\log(P(\mathbf{y}|\mathbf{f},\mathbf{I},\theta_n))\} + E_{\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}}\{\log(P(\mathbf{f}|\mathbf{X},\theta_{GP}))\} \\ & + \log(P(\mathbf{I}|\theta_n))\Big\} \end{aligned} \tag{2.20}$$

By substituting Eq.(2.8) and Eq.(2.10) into Eq.(2.20), the $Q(\vartheta;\vartheta^{(t)})$ function is

derived as,

$$
\begin{aligned}
Q(\vartheta;\vartheta^{(t)}) &= E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Bigg\{E_{\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}}\Big\{-\frac{n}{2}\log 2\pi - \log|diag(\sigma_{\mathbf{I}})| \\
&\quad - \frac{(\mathbf{y}-\mathbf{f})^T\Big(diag(\sigma_{\mathbf{I}})\Big)^{-2}(\mathbf{y}-\mathbf{f})}{2}\Big\} + E_{\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}}\Big\{-\frac{1}{2}\log|2\pi\mathbf{K}| \\
&\quad - \frac{(\mathbf{f}-m\mathbf{1})^T(\mathbf{K})^{-1}(\mathbf{f}-m\mathbf{1})}{2}\Big\} + \log(P(\mathbf{I}|\theta_n))\Bigg\} \\
&= E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Bigg\{-\frac{n}{2}log2\pi - \log|diag(\sigma_{\mathbf{I}})| - \frac{1}{2}\log|2\pi\mathbf{K}| \\
&\quad - E_{\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}}\Big\{\frac{(\mathbf{y}-\mathbf{f})^T\Big(diag(\sigma_{\mathbf{I}})\Big)^{-2}(\mathbf{y}-\mathbf{f})}{2}\Big\} \\
&\quad - E_{\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}}\Big\{\frac{(\mathbf{f}-m\mathbf{1})^T(\mathbf{K})^{-1}(\mathbf{f}-m\mathbf{1})}{2}\Big\} + \log(P(\mathbf{I}|\theta_n))\Bigg\}
\end{aligned}
\tag{2.21}
$$

To find the expected value of Q function with respect to $\mathbf{f}$, the posterior distributions, $P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)})$ and $P(\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)})$, need to be derived. Hence, we now proceed to derive $P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)})$ according to Bayes' rule,

$$
P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}) = \frac{P(\mathbf{y}|\mathbf{f},\mathbf{I},\theta_n^{(t)})P(\mathbf{f}|\mathbf{X},\theta_{GP}^{(t)})}{P(\mathbf{y}|\mathbf{I},\mathbf{X},\vartheta^{(t)})} \tag{2.22}
$$

where the denominator in Eq.(2.22) is the normalizing constant, which is independent of the function values. As a result, the posterior distribution over $\mathbf{f}$, will only have the functional dependencies on the likelihood and the prior, as follows,

$$
P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}) \sim P(\mathbf{y}|\mathbf{f},\mathbf{I},\theta_{\mathbf{n}}^{(\mathbf{t})})P(\mathbf{f}|\mathbf{X},\theta_{GP}^{(t)}) \tag{2.23}
$$

As we assume mixtures of Gaussian noise profile, the probability distribution of the observation given hidden noise mode identity is given as,

$$
P(\mathbf{y}|\mathbf{f},\mathbf{I},\theta_{\mathbf{n}}^{(\mathbf{t})})) \sim \mathcal{N}(\mathbf{y}|\mathbf{f},\Big(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\Big)^2) \tag{2.24}
$$

where $\sigma_I = [\sigma_{I_1}\sigma_{I_2}...\sigma_{I_n}]$ represents noise variances for n observations. For a mixture of two Gaussian noise components, $\sigma_{I_i} \in [\sigma_1,\sigma_2]$. Since the likelihood and the prior are both Gaussian, the posterior over $\mathbf{f}$ is tractable and can be determined analytically. By performing completing the square method, we obtain,

$$P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}) \sim \mathcal{N}(\mathbf{y}|\mathbf{f},\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^2)\mathcal{N}(\mathbf{f}|m^{(t)}\mathbf{1},\mathbf{K}^{(\mathbf{t})}) \tag{2.25}$$

$$\sim \exp\Bigg[-\frac{1}{2}\Bigg((\mathbf{y}-\mathbf{f})^T\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}(\mathbf{y}-\mathbf{f}) + (\mathbf{f}-m^{(t)}\mathbf{1})^T\mathbf{K}^{(\mathbf{t})-\mathbf{1}}(\mathbf{f}-m^{(t)}\mathbf{1})\Bigg)\Bigg] \tag{2.26}$$

$$\sim \exp\Bigg[-\frac{1}{2}\Bigg(\mathbf{y}^{\mathbf{T}}\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}\mathbf{y} + \mathbf{f}^{\mathbf{T}}\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}\mathbf{f} - \mathbf{y}^{\mathbf{T}}\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}\mathbf{f} - \mathbf{f}^{\mathbf{T}}\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}\mathbf{y} + \mathbf{f}^{\mathbf{T}}\mathbf{K}^{(\mathbf{t})-1}\mathbf{f} - m^{(t)}\mathbf{1}^{\mathbf{T}}\mathbf{K}^{(\mathbf{t})-1}\mathbf{f} - \mathbf{f}^{\mathbf{T}}\mathbf{K}^{(\mathbf{t})-1}m\mathbf{1} + m^{(t)}\mathbf{1}^{\mathbf{T}}\mathbf{K}^{(\mathbf{t})-1}m^{(t)}\mathbf{1}\Bigg)\Bigg] \tag{2.27}$$

$$\sim \exp\Bigg[-\frac{1}{2}\Bigg(\mathbf{f}^{\mathbf{T}}\left(\mathbf{K}^{(\mathbf{t})-1} + \left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}\right)\mathbf{f} - \mathbf{f}^{\mathbf{T}}\left(\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}\mathbf{y} + \mathbf{K}^{(\mathbf{t})-1}m^{(t)}\mathbf{1}\right) - \left(\mathbf{y}^{\mathbf{T}}\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2} + m^{(t)}\mathbf{1}^{\mathbf{T}}\mathbf{K}^{(\mathbf{t})-1}\right)\mathbf{f} + \underbrace{\mathbf{y}^{\mathbf{T}}\left(diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})\right)^{-2}\mathbf{y} + m^{(t)}\mathbf{1}^{\mathbf{T}}\mathbf{K}^{(\mathbf{t})-1}m^{(t)}\mathbf{1}}\Bigg)\Bigg] \tag{2.28}$$

After reorganizing all terms, we drop terms in the under-brace in Eq.(2.28), which do not involve $\mathbf{f}$. Letting $A = (\mathbf{K}^{(\mathbf{t})-1} + diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})^{-2})$ and $B = (diag(\sigma_{\mathbf{I}}^{(\mathbf{t})})^{-2}\mathbf{y} + \mathbf{K}^{(\mathbf{t})-1}m^{(t)}\mathbf{1})$, Eq.(2.28) becomes,

$$P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}) \sim \exp[-\frac{1}{2}(\mathbf{f}^{\mathbf{T}}\mathbf{A}\mathbf{f} - \mathbf{f}^{\mathbf{T}}\mathbf{B} - \mathbf{B}^{\mathbf{T}}\mathbf{f})] \tag{2.29}$$

By doing some straightforward algebraic manipulations on Eq.(2.29), the following expression can be obtained,

$$P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}) \sim \exp[-\frac{1}{2}(\mathbf{f}-\mathbf{A}^{-\mathbf{1}}\mathbf{B})^T\mathbf{A}(\mathbf{f}-\mathbf{A}^{-\mathbf{1}}\mathbf{B})] \tag{2.30}$$

We now introduce new notation for the compact representation of the equations. Let $\mathbf{\Sigma_n} = \mathbf{A^{-1}}$ and $\mu_{\mathbf{n}} = \mathbf{A^{-1}B}$, respectively, be the covariance and the mean of $\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}$. As a result Eq.(2.30) can be rewritten as below,

$$P(\mathbf{f}|\mathbf{y},\mathbf{I},\mathbf{X},\vartheta^{(t)}) \sim \exp[-\frac{1}{2}(\mathbf{f}-\mu_{\mathbf{n}}{}^{(t)})^T\mathbf{\Sigma_n}^{(\mathbf{t})-\mathbf{1}}(\mathbf{f}-\mu_{\mathbf{n}}{}^{(t)})] \tag{2.31}$$

where $\mathbf{\Sigma_n}^{(\mathbf{t})} = \left(\mathbf{K}^{(\mathbf{t})^{-1}} + \left(diag(\sigma_\mathbf{I}^{(\mathbf{t})})\right)^{-2}\right)^{-1}$ and $\mu_\mathbf{n}^{(\mathbf{t})} = \left(\mathbf{K}^{(\mathbf{t})^{-1}} + \left(diag(\sigma_\mathbf{I}^{(\mathbf{t})})\right)^{-2}\right)^{-1} * \left(\left(diag(\sigma_\mathbf{I}^{(\mathbf{t})})\right)^{-2}\mathbf{y} + \mathbf{K}^{(\mathbf{t})^{-1}} m^{(t)}\mathbf{1}\right)$, resulting the expectation of the quadratic form and having the posterior distribution of $P(\mathbf{f}|\mathbf{y}, \mathbf{I}, \mathbf{X}, \vartheta^{(t)}) \sim \mathcal{N}(\mathbf{f}|\mu_\mathbf{n}^{(\mathbf{t})}, \mathbf{\Sigma_n}^{(\mathbf{t})})$.

The posterior distribution $P(\mathbf{I}|\mathbf{y}, \mathbf{X}, \vartheta^{(t)})$ is now derived using Bayes' rule to get the complete form of Q function,

$$P(\mathbf{I}|\mathbf{y}, \mathbf{X}, \vartheta^{(t)}) = \frac{P(\mathbf{y}|\mathbf{I}, \mathbf{X}, \vartheta^{(t)})P(\mathbf{I}|\theta_n^{(t)})}{\sum_I P(\mathbf{y}|\mathbf{I}, \mathbf{X}, \vartheta^{(t)})P(\mathbf{I}|\theta_n^{(t)})} \tag{2.32}$$

where the conditional probability of $P(\mathbf{y}|\mathbf{I}, \mathbf{X}, \vartheta^{(t)})$ can be obtained through marginalization of Eq.(2.33) over $\mathbf{f}$,

$$P(\mathbf{y}|\mathbf{I}, \mathbf{X}, \vartheta^{(t)}) = \int P(\mathbf{y}|\mathbf{f}, \mathbf{I}, \mathbf{X}, \theta_n^{(t)})P(\mathbf{f}|\mathbf{X}, \theta_{GP}^{(t)})\ d\mathbf{f} \tag{2.33}$$

Eq.(2.33) is further simplified using the properties of Gaussian integrals (A.2),

$$P(\mathbf{y}|\mathbf{I}, \mathbf{X}, \vartheta^{(t)}) = \int \mathcal{N}(\mathbf{y}|\mathbf{f}, \left(diag(\sigma_\mathbf{I})\right)^2)\mathcal{N}(\mathbf{f}|m\mathbf{1}, \mathbf{K})\ d\mathbf{f} \tag{2.34}$$

$$= \mathcal{N}(\mathbf{y}|m\mathbf{1}, \left(diag(\sigma_\mathbf{I})\right)^2 + \mathbf{K}) \tag{2.35}$$

Since there are 2^n total number of possible combinations to be enumerated due to n samples and 2 components for the calculation of denominator of posterior probability of $P(\mathbf{I}|\mathbf{y}, \mathbf{X}, \vartheta^{(t)})$, we need to do some approximations to avoid combinatorial solutions. First, we are approximating the possibility of occurrence of the noise mode identity I_i as random and does not depend on the occurrence of other model identities $I_j,\ j \neq i$, which leads to following approximation for $P(\mathbf{I}|\mathbf{y}, \mathbf{X}, \vartheta^{(t)})$ as,

$$P(\mathbf{I}|\mathbf{y}, \mathbf{X}, \vartheta^{(t)}) = \prod_{i=1}^{n} P(I_i|y_i, \mathbf{x_i}, \vartheta^{(t)}) \tag{2.36}$$

Further, the posterior probability of $P(I_i|y_i, \mathbf{x_i}, \vartheta^{(t)})$ can be written as,

$$\begin{aligned} P(I_i = j|y_i, \mathbf{x_i}, \vartheta^{(t)}) &= \frac{P(y_i|I_i = j, x_i, \vartheta^{(t)})P(I_i = j|\theta_n^{(t)})}{\sum_{j=1}^{2} P(y_i|I_i = j, x_i, \vartheta^{(t)})P(I_i = j|\theta_n^{(t)})} \\ &= \frac{\frac{1}{\sqrt{2\pi\sigma_j^{(t)^2}}} \exp[-\frac{(y_i - f_i^{(t)})^2}{2\sigma_j^{(t)^2}}]\ \alpha_j^{(t)}}{\sum_{j=1}^{2} \frac{1}{\sqrt{2\pi\sigma_j^{(t)^2}}} \exp[-\frac{(y_i - f_i^{(t)})^2}{2\sigma_j^{(t)^2}}]\ \alpha_j^{(t)}} \triangleq \gamma_{ij}^{(t)} \end{aligned} \tag{2.37}$$

While deriving Eq.(2.37), we have also employed the following approximation: $P(y|\mathbf{x}, I, \vartheta^{(t)})$ follows univariate Gaussian, and the deterministic part of GP, i.e. $f_i^{(t)}$ can be replaced with μ_{n_i}, from the vector $\mu_n^{(t-1)} = [\mu_{n_1}^{(t-1)}, \mu_{n_2}^{(t-1)}, \ldots, \mu_{n_i}^{(t-1)}, \ldots, \mu_{n_n}^{(t-1)}]$ which is the mean value of posterior probability of $P(\mathbf{f}|\mathbf{y}, \mathbf{I}, \mathbf{X}, \vartheta^{(t-1)})$ obtained from the last iteration in the E-step.

Using the result of Eq.(2.31) in the Q function expression Eq.(2.21), we obtain,

$$\begin{aligned} Q(\vartheta; \vartheta^{(t)}) = & -\frac{n}{2}\log 2\pi - E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{\log|diag(\sigma_{\mathbf{I}})|\Big\} - \frac{1}{2}\log|2\pi\mathbf{K}| \\ & - \frac{1}{2}E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{Tr\Big(\big(diag(\sigma_{\mathbf{I}})\big)^{-2}\boldsymbol{\Sigma}_{\mathbf{n}}^{(\mathbf{t})}\Big)\Big\} \\ & - \frac{1}{2}E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{(\mathbf{y}-\mu_{\mathbf{n}}^{(\mathbf{t})})^T\big(diag(\sigma_{\mathbf{I}})\big)^{-2}(\mathbf{y}-\mu_{\mathbf{n}}^{(\mathbf{t})})\Big\} \\ & - \frac{Tr(\mathbf{K}^{-1}\boldsymbol{\Sigma}_{\mathbf{n}}^{(\mathbf{t})}) + (\mu_{\mathbf{n}}^{(\mathbf{t})} - m\mathbf{1})^T\mathbf{K}^{-1}(\mu_{\mathbf{n}}^{(\mathbf{t})} - m\mathbf{1})}{2} \\ & + E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{\log(P(\mathbf{I}|\theta_n))\Big\} \end{aligned} \tag{2.38}$$

Enumerating each expectation term with respect to $\mathbf{I}$ in the Q function (2.38) yields:

$$\begin{aligned} E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{\log|diag(\sigma_{\mathbf{I}})|\Big\} &= E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{\sum_{i=1}^{n}\log\sigma_{I_i}\Big\} \\ &= \sum_{j=1}^{2}\sum_{i=1}^{n}P(I_i = j|y_i, \mathbf{x_i}, \vartheta^{(t)})\log\sigma_{I_i=j} \end{aligned} \tag{2.39}$$

$$\begin{aligned} E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{Tr\Big(\big(diag(\sigma_{\mathbf{I}})\big)^{-2}\boldsymbol{\Sigma}_{\mathbf{n}}^{(\mathbf{t})}\Big)\Big\} &= E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{\sum_{i=1}^{n}\frac{\Sigma_n{}_{ii}^{(t)}}{\sigma_{I_i}^2}\Big\} \\ &= \sum_{j=1}^{2}\sum_{i=1}^{n}P(I_i = j|y_i, \mathbf{x_i}, \vartheta^{(t)})\left(\frac{\Sigma_n{}_{ii}^{(t)}}{\sigma_{I_i=j}^2}\right) \end{aligned} \tag{2.40}$$

$$\begin{aligned} E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{(\mathbf{y}-\mu_{\mathbf{n}}^{(\mathbf{t})})^T\big(diag(\sigma_{\mathbf{I}})\big)^{-2}(\mathbf{y}-\mu_{\mathbf{n}}^{(\mathbf{t})})\Big\} &= E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\Big\{\frac{(y_i-\mu_{n_i}^{(t)})^2}{\sigma_{I_i}^2}\Big\} \\ &= \sum_{j=1}^{2}\sum_{i=1}^{n}P(I_i = j|y_i, \mathbf{x_i}, \vartheta^{(t)})\left(\frac{(y_i-\mu_{n_i}^{(t)})^2}{\sigma_{I_i=j}^2}\right) \end{aligned} \tag{2.41}$$

$$E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\left\{\log\Big(P(\mathbf{I}|\theta_n)\Big)\right\} = E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\left\{\log\Big(P(I_i|\theta_n)\Big)\right\} = E_{\mathbf{I}|\mathbf{y},\mathbf{X},\vartheta^{(t)}}\left\{\log(\alpha_i)\right\}$$
$$= \sum_{j=1}^{2}\sum_{i=1}^{n} P(I_i = j|y_i, \mathbf{x_i}, \vartheta^{(t)}) \log \alpha_{I_i=j} \tag{2.42}$$

Finally, by deploying Eqs.(3.27)-(2.42) on Eq.(2.38), the Q function is derived as,

$$\begin{aligned} Q(\vartheta;\vartheta^{(t)}) = &-\frac{n}{2}\log 2\pi - \sum_{j=1}^{2}\sum_{i=1}^{n}\gamma_{ij}^{(t)}\log\sigma_{I_i=j} - \frac{1}{2}\log|2\pi\mathbf{K}| \\ &- \frac{1}{2}\sum_{j=1}^{2}\sum_{i=1}^{n}\gamma_{ij}^{(t)}\left(\frac{\Sigma_{n}{}_{ii}^{(t)}}{\sigma^2_{I_i=j}}\right) \\ &- \frac{1}{2}\sum_{j=1}^{2}\sum_{i=1}^{n}\gamma_{ij}^{(t)}\left(\frac{(y_i-\mu_{n_i}^{(t)})^2}{\sigma^2_{I_i=j}}\right) \\ &- \frac{Tr(\mathbf{K}^{-1}\mathbf{\Sigma_n^{(t)}}) + (\mu_{\mathbf{n}}^{(\mathbf{t})} - m\mathbf{1})^T(\mathbf{K})^{-1}(\mu_{\mathbf{n}}^{(\mathbf{t})} - m\mathbf{1})}{2} \\ &+ \sum_{j=1}^{2}\sum_{i=1}^{n}\gamma_{ij}^{(t)}\log\alpha_{I_i=j} \end{aligned} \tag{2.43}$$

As we have now derived the Q function, next subsection illustrates the M-step, where the derivations of the parameter update expressions are carried out.

2.4.2 M-Step

In the M-step, we maximize the Q function with respect to ϑ, that is, θ_{GP} and θ_n, respectively, to obtain the updated estimate of the parameters. We obtain the following expressions by calculating the gradient of Q function with respect to GP hyper-parameters ($\theta_{GP} = [\theta_{cov}, \theta_{mean}]$),

$$\begin{aligned} \frac{\partial Q(\vartheta;\vartheta^{(t)})}{\partial\theta_{cov}} =& -\frac{1}{2}tr(\mathbf{K^{-1}}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}) + \frac{1}{2}(\boldsymbol{\mu_n}^{(\mathbf{t})} - m\mathbf{1})^T\mathbf{K^{-1}}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\mathbf{K^{-1}}(\boldsymbol{\mu_n}^{(\mathbf{t})} - m\mathbf{1}) \\ &+ \frac{1}{2}tr(\mathbf{K^{-1}}\frac{\partial\mathbf{K}}{\partial\theta_{cov}}\mathbf{K}^{-1}\mathbf{\Sigma_n}^{(\mathbf{t})}) \\ \frac{\partial Q(\vartheta;\vartheta^{(t)})}{\partial\theta_{mean}} =& \mathbf{1}^T\mathbf{K^{-1}}(\boldsymbol{\mu_n}^{(\mathbf{t})} - m\mathbf{1}) \end{aligned} \tag{2.44}$$

As the above expression does not have a closed form solution, we resort to numerical solutions. Solutions of Eq.(3.31) by gradients descent method provide the value

of θ_{GP} that maximizes the Q function. By setting the derivative of Q function with respect to the noise hyper-parameters to zero and solving for parameter values, the update equation for the variances of noise components, $\sigma_p^2 \in [\sigma_1^2, \sigma_2^2]$ is obtained as,

$$\frac{\partial Q(\vartheta; \vartheta^{(t)})}{\partial \sigma_p^2} = 0 \Rightarrow {\sigma_p^2}^{(t+1)} = \frac{\sum_{i=1}^{n} {\gamma_{ip}}^{(t)} ({\Sigma_{n_{ii}}}^{(t)} + (y_i - {\mu_{n_i}}^{(t)})^2)}{\sum_{i=1}^{n} {\gamma_{ip}}^{(t)}} \tag{2.45}$$

Maximization of the Q function to calculate α_p becomes a constrained optimization problem, where $\sum_{j=1}^{2} \alpha_j = 1$ is the constraint that should be satisfied. To solve this constrained optimization problem, the Lagrangian multiplier should be introduced, and the derivative of Lagrangian expression over α_p and Lagrange multiplier is set to be zero [Vapnyarskii, 2001]. We omit the detail steps for brevity and the update equation for α_p is given as,

$${\alpha_p}^{(t+1)} = \frac{\sum_{i=1}^{n} {\gamma_{ip}}^{(t)}}{n} \tag{2.46}$$

This completes the M-step, as we have derived the update expressions for all the parameters. The E-step and M-step are solved in iterative manner, until the converged parameters are obtained.

2.5 Prediction with proposed GPR model

After convergence of EM algorithm, the parameter estimates of the proposed GPR model are obtained, which can be further employed for prediction. To make predictions for given test data, we compute the conditional distribution of function values $\mathbf{f}_+$ corresponding to test input data $\mathbf{X}_+$. To compute the posterior predictive distribution of $\mathbf{f}_+|\mathbf{y}$, we need to first calculate the joint distribution of $\mathbf{f}_+, \mathbf{y}$ as,

$$P(\mathbf{y}, \mathbf{f}_+|\mathbf{X}, \mathbf{X}_+, \vartheta) \sim \mathcal{N}(\begin{bmatrix} m\mathbf{1} \\ m_+\mathbf{1} \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \left(\mathbf{diag}(\sigma_\mathbf{I})\right)^\mathbf{2} & \mathbf{K}(\mathbf{X}, \mathbf{X}_+) \\ \mathbf{K}(\mathbf{X}_+, \mathbf{X}) & \mathbf{K}(\mathbf{X}_+, \mathbf{X}_+) \end{bmatrix}) \tag{2.47}$$

where $\left(diag(\sigma_\mathbf{I})\right)^{-2}$ is the estimated noise covariance, $\mathbf{K}(\mathbf{X}_+, \mathbf{X}_+)$ is the prior covariance matrix of $\mathbf{f}_+$, and $\mathbf{K}(\mathbf{X}, \mathbf{X}_+)$ is the covariance between $\mathbf{f}_+$ and $\mathbf{y}$. Based on the results provided in A.1, the of conditional distribution $P(\mathbf{f}_+|\mathbf{y}, \mathbf{X}, \mathbf{X}_+, \vartheta)$ can be

derived as,

$$
\begin{aligned}
&P(\mathbf{f}_+|\mathbf{y},\mathbf{X},\mathbf{X}_+,\vartheta) \sim \mathcal{N}(\boldsymbol{n}_+,\mathbf{S}_+) \\
&\quad where \qquad \boldsymbol{n}_+ = m_+\mathbf{1} + \mathbf{K}(\mathbf{X}_+,\mathbf{X})\Big(\mathbf{K}(\mathbf{X},\mathbf{X}) + \Big(\mathbf{diag}(\sigma_{\mathbf{I}})\Big)^{\mathbf{2}}\Big)^{-1}(\mathbf{y} - m\mathbf{1}), \\
&\qquad\qquad\quad \mathbf{S}_+ = \mathbf{K}(\mathbf{X}_+,\mathbf{X}_+) - \mathbf{K}(\mathbf{X}_+,\mathbf{X})\Big(\mathbf{K}(\mathbf{X},\mathbf{X}) + \Big(\mathbf{diag}(\sigma_{\mathbf{I}})\Big)^{\mathbf{2}}\Big)^{-1}\mathbf{K}(\mathbf{X},\mathbf{X}_+)
\end{aligned}
\tag{2.48}
$$

2.6 Algorithm

Fig.(2.3) presents the flow-chart of the proposed GPR model parameter estimation for prediction. The first step of the algorithm is to set the initial value for both Gaussian process hyper-parameters and noise likelihood hyper-parameters. Further, a standard GP is used to train the model from the training dataset and the predictive mean thus obtained, is set as an initial value for $\mathbf{f}$. In the E-step, the posterior probability of $P(\mathbf{f}|\mathbf{y},\mathbf{X},\mathbf{I})$ and $P(\mathbf{I}|\mathbf{y},\mathbf{X})$ is inferred. Then, noise mode identity vector is constructed by comparing the probability in Eq.(3.27) for each component, and the component that has the largest probability will be chosen as a noise mode identity of that sample. In the M-step, using the posteriors obtained in the E-step, we maximize the Q function as a function of $\vartheta^{(t)}$, and the hyper-parameters are updated using Eq.(3.31) to Eq.(3.32). The E-step and M-step are alternated until convergence. In the last step of the algorithm, the GPR with the trained hyper-parameters will be used to predict response for a given test data using Eq.(3.46).

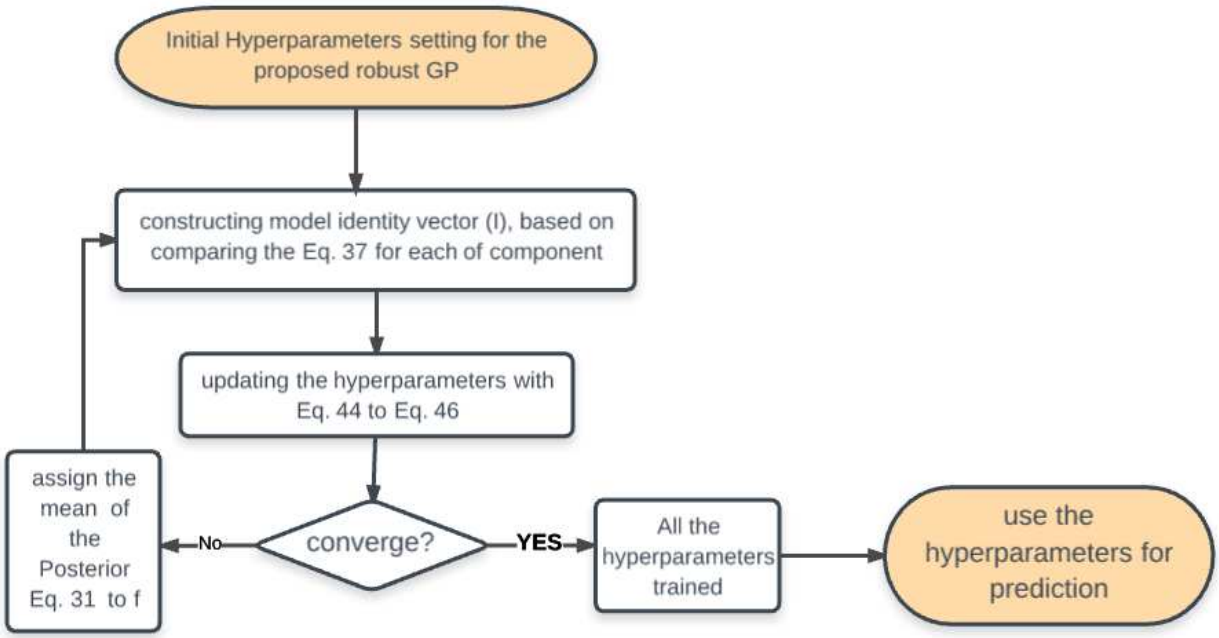

Figure 2.3: Algorithm of the Proposed robust Gaussian process

2.7 Examples

To evaluate the efficacy of the presented GPR in this chapter, we provide simulation examples for various cases, namely, (i) two synthetic data sets with one dimensional and multidimensional inputs, respectively, (ii) a simulation example using continuous stirred tank reactor (CSTR) and (iii) an industrial example. To statistically characterize the performance, we employ three metrics, namely, mean absolute error (MAE), root mean square error (RMSE), and negative log of predictive probability (NLP) as formulated below [Kuss, 2006]:

- Mean absolute error:

$$MAE = \frac{\sum_{i=1}^{N_+} |\hat{f}_{+_i} - f_{+_i}|}{N_+} \tag{2.49}$$

- Root mean square error:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_+} (\hat{f}_{+_i} - f_{+_i})^2}{N_+}} \tag{2.50}$$

- Negative logarithm of predictive probability

$$NLP = -\frac{\sum_{i=1}^{N_+} \log P(\mathbf{f}_{+_\mathbf{i}} | \mathbf{y}, \mathbf{X}, \mathbf{x}_+)}{N_+} \tag{2.51}$$

where $\hat{f}_+$ and f_+ are the predicted value and the reference respectively while N denotes the number of samples in test data set.

2.7.1 Numerical Example

Neal example

As a first example, one dimensional problem is considered, which is adopted from Neal [1997]. The input signal for the training part, x, was drawn from a Gaussian distribution with zero mean and unit variance, and the corresponding target value was calculated using the function given below,

$$f(x) = 0.3 + 0.4x + 0.5\sin(2.7x) + \frac{1.1}{1+x^2} \tag{2.52}$$

We injected measurement noise to the target value where noise was drawn from a Gaussian mixture model where 85% of the noise realizations were generated from $\mathcal{N}(0, 0.01)$, and the remaining 15% of the noise realizations were generated from $\mathcal{N}(0, 1.5)$. In our example, training data set contains 100 samples, and test data set contains 1000 samples. The data set was modelled using the proposed method and the results are compared with other models in the existing literature such as, standard GP with Gaussian noise, GP with Student's t-distribution likelihood, and GP with Laplace noise distribution. We use standard GP to refer to GPR with single Gaussian noise whose parameters are estimated by maximum likelihood estimation, and GPL via VB and GPT via VB stand for GPR with Laplace noise and with student-t noise, respectively, which are implemented using Variational Bayesian inference simulated using GPML toolbox [Rasmussen and Nickisch, 2010]. GPM via EM denotes the proposed method of this chapter.

We have used SE kernel function, which is formulated for one dimensional input as below,

$$k_{SE}(x_i, x_j) = \sigma_f^2 \ \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) \tag{2.53}$$

The 100 training cases and the result of cross validation of these four models on the test data set are presented in Fig.(2.4). Table 2.2 shows the RMSE, MAE, and NLP of four different regression models. The following are the set of hyper-parameters computed by the proposed method:

$$\vartheta = [\log(l), \sigma_f, \alpha_1, \alpha_2, \sigma_1^2, \sigma_2^2] = [0.9439, 1.3928, 0.8761, 0.1239, 0.0074, 1.5521] \tag{2.54}$$

Table 2.2: Prediction Performance of Neal

Method	MAE	RMSE	NLP
Standard GP	0.2069	0.3006	0.3412
Robust GP with a mixture noise	**0.0420**	**0.0528**	**−1.7488**
Robust GP with T-student likelihood	0.1975	0.4722	-0.3668
Robust GP with Laplace likelihood	0.1763	0.411	-0.1278

As it is evident from the results that the proposed method outperforms the other

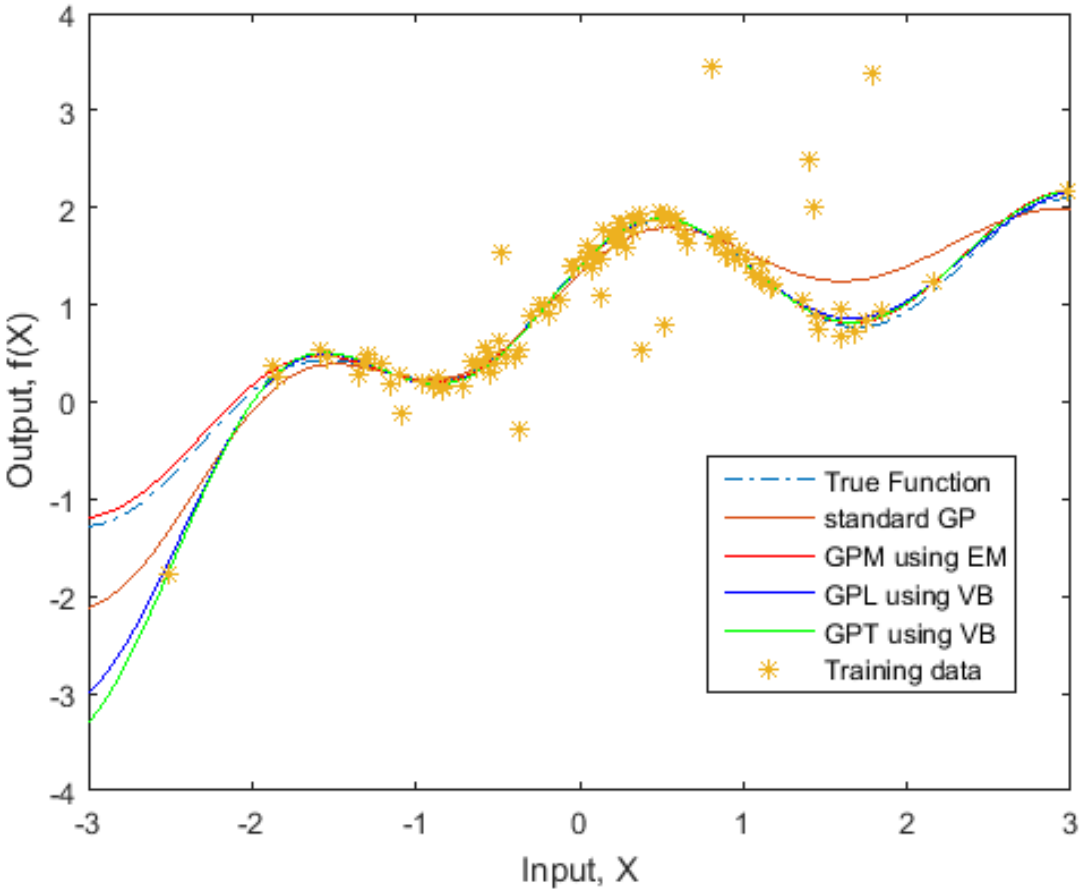

Figure 2.4: The predictive mean using the four models on Neal example

methods in terms of prediction performance while the estimated noise likelihood parameters are also very much close to the true values.

Furthermore, we investigate how the proposed method works in this example if the injected noise was drawn from a single Gaussian noise with zero mean and variance 0.1. The Table 2.3 illustrates the estimated parameters using the proposed method and Standard GP.

Table 2.3: Estimated parameters for Neal example with single Gaussian noise

parameters / Method	Variances		Mixture weights	
	σ_1	σ_2	α_1	α_2
True value	0.1	NA	1	0
Standard GP	0.1070	NA	1	0
GPGMM	0.1102	0.1008	0.5564	0.4436

Using Fig. (2.5) and Table 2.4, we can compare the results from prediction performance perspective for the case with single Gaussian noise. As it is evident from the results, the proposed method has better prediction performance even in the case with single Gaussian noise.

Table 2.4: Prediction Performance of Neal example with single Gaussian noise

Method	MAE	RMSE
Standard GP	0.1361	0.2280
Robust GP with mixture noise	0.1358	0.2276

Friedman data set

Friedman [1991] described the following nonlinear model in Eq.(2.55) that contains 10 covariates in the data ($x = x_1, x_2, ..., x_{10}$). However, the function that describes the response $f(\mathbf{x})$ is dependent only on the first five dimensions (x_1 to x_5),

$$f(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \tag{2.55}$$

10 training data sets of size of 100×10 are generated as 100 by 10 array of random numbers between 0 and 1 sampled from continuous uniform distributions. After obtaining the corresponding function values from this model, we normalize data

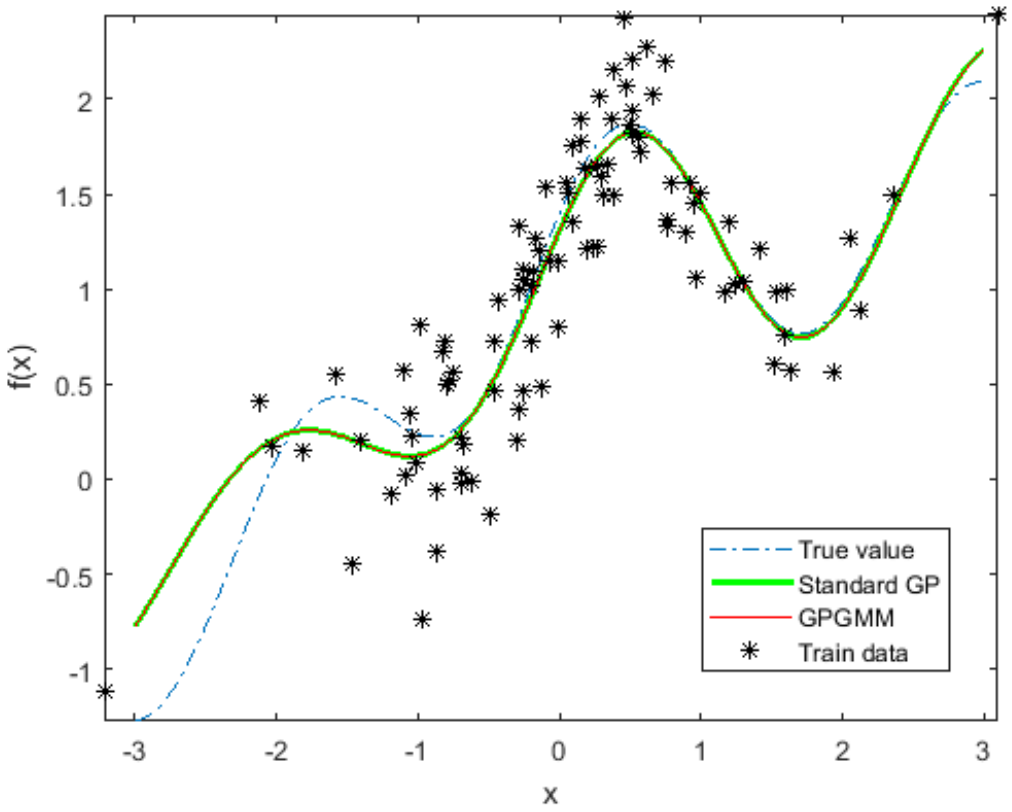

Figure 2.5: The mean prediction on Neal example with single Gaussian noise

set, and the noise from a mixture of two Gaussians is added. For generating noise, we assumed that 70 percent of noise realizations comes from $\mathcal{N}(0, 0.002)$ and the remaining noise realizations are sampled from $\mathcal{N}(0, 1.08)$. To evaluate the proposed model, we generate a test data set of 10000 function values from the Friedman model. We use the SE kernel function with 10 dimensional inputs,

$$k_{SE}(\mathbf{x_i}, \mathbf{x_j}) = \sigma_f^2 \ \exp\Big[\sum_{d=1}^{10}\Big(-\frac{(x_{id} - x_{jd})^2}{2l_d^2}\Big)\Big] \tag{2.56}$$

Fig.(2.6) and Fig.(2.7) show MAE and RMSE on the 10 different data sets where we observe that the proposed model performs better than the other models and gives more accurate prediction. For further comparison, the scatter plot of the four model predictions is also provided based on one of the training sets in Fig.(2.8). The estimated GPR parameters values for the data are given below:

$$\begin{aligned}\vartheta &= [\log(l_1), \log(l_2), \log(l_3), \ldots, \log(l_9), \log(l_{10}), \sigma_f, \alpha_1, \alpha_2, \sigma_1^2, \sigma_2^2] \\ &= [0.4284, -0.7027, -0.1624, 0.5143, 1.2521, 10.5515, 10.4032, 10.5210, 2.8724, 9.4514, \\ &\qquad 1.6725, 0.2506, 0.7494, 1.0849, 0.0143]\end{aligned} \tag{2.57}$$

where the log scale values are large for the inputs from $x_6, \ldots, x_{10}$ which shows the proposed approach can qualitatively describe the relevant characteristics of Eq.(2.55).

Furthermore, we can also observe that the estimated parameters for the noise are very close to the true noise parameters.

2.7.2 Continuous stirred tank reactor

The reactor considered here is a CSTR with an irreversible and exothermic reaction $A \to B$ having the mass and energy Balances as follows [Morningred et al., 1990],

$$\frac{dC_A(t)}{dt} = \frac{F}{V}(C_{Af}(t) - C_A(t)) - k_0 C_A(t) \exp(-\frac{E}{RT(t)}) \tag{2.58}$$

$$\begin{aligned}\frac{dT(t)}{dt} =& \frac{F}{V}(T_f(t) - T(t)) - \frac{(-\Delta H)k_0 C_A(t)}{\rho C_p} \exp(-\frac{E}{RT(t)}) \\ &- \frac{\rho_c C_{pc}}{\rho C_p V} F_c(t)\Big\{1 - \exp[\frac{-hA}{F_c(t)\rho_c C_{pc}}]\Big\}(T(t) - T_j(t))\end{aligned} \tag{2.59}$$

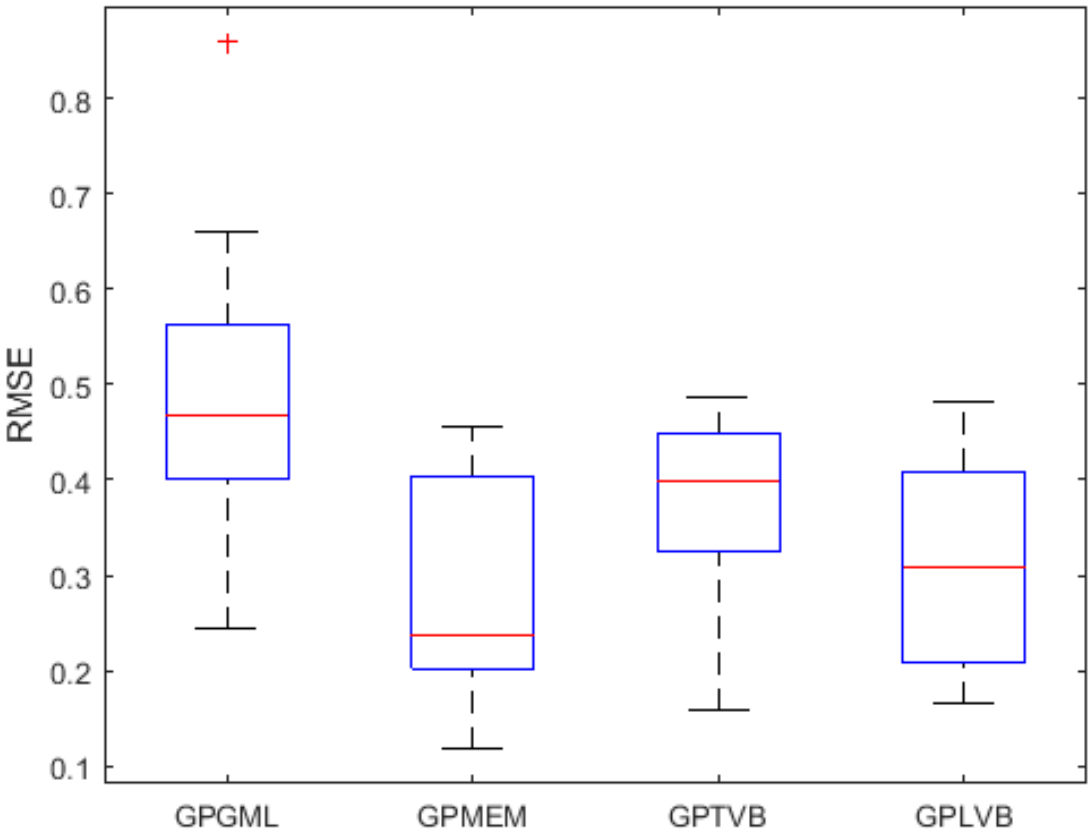

Figure 2.6: Box plot of the RMSE on the 10 data sets

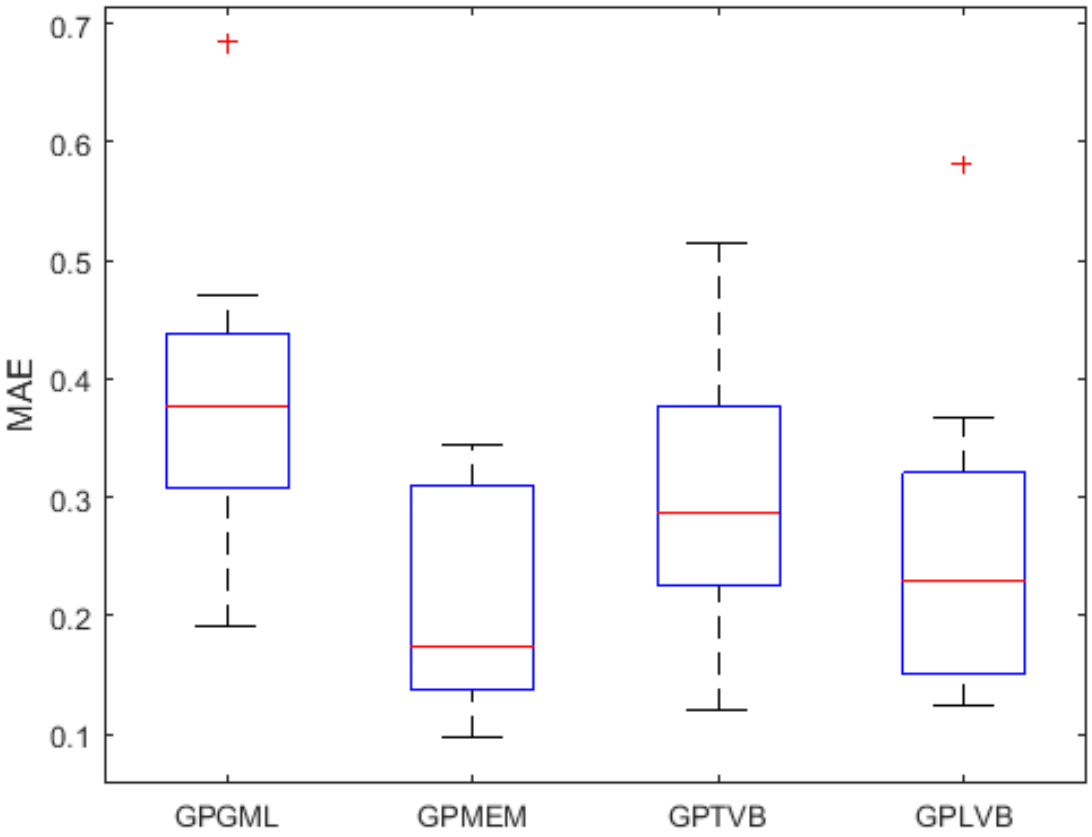

Figure 2.7: Box plot of the MAE on the 10 data sets

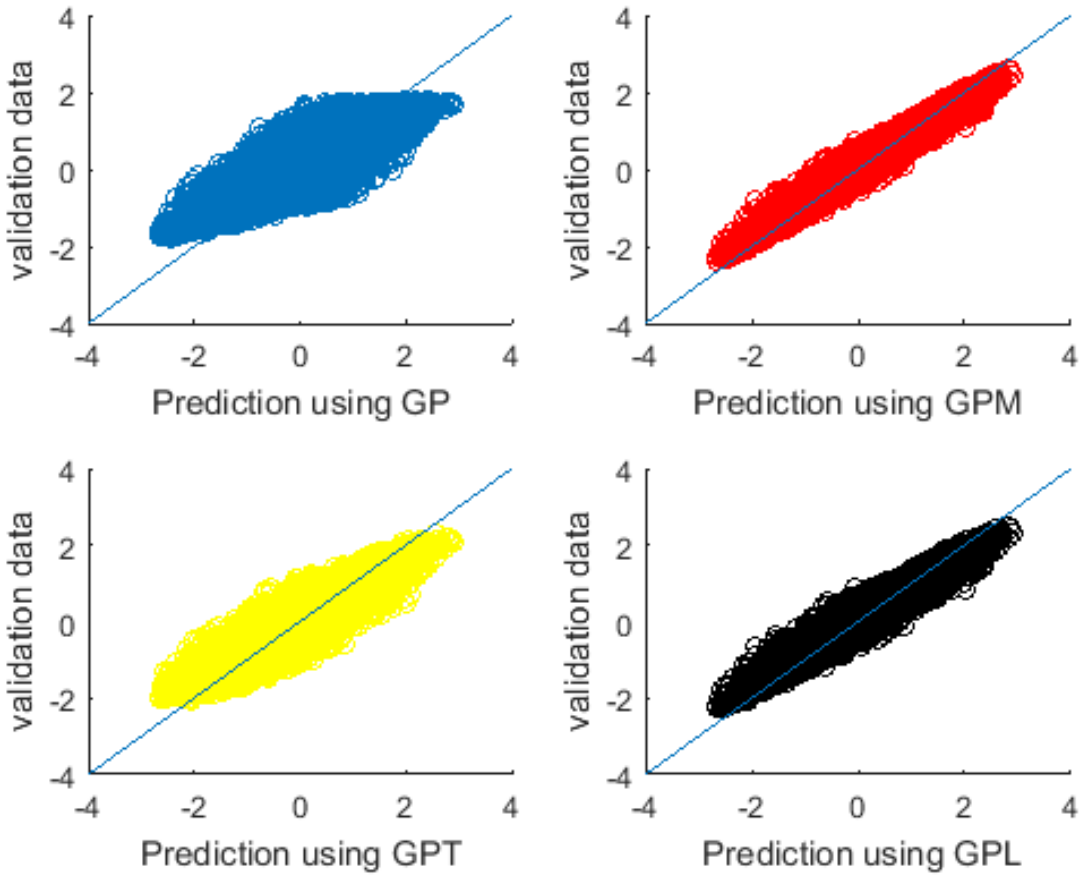

Figure 2.8: Scatter Plot on one of the data sets

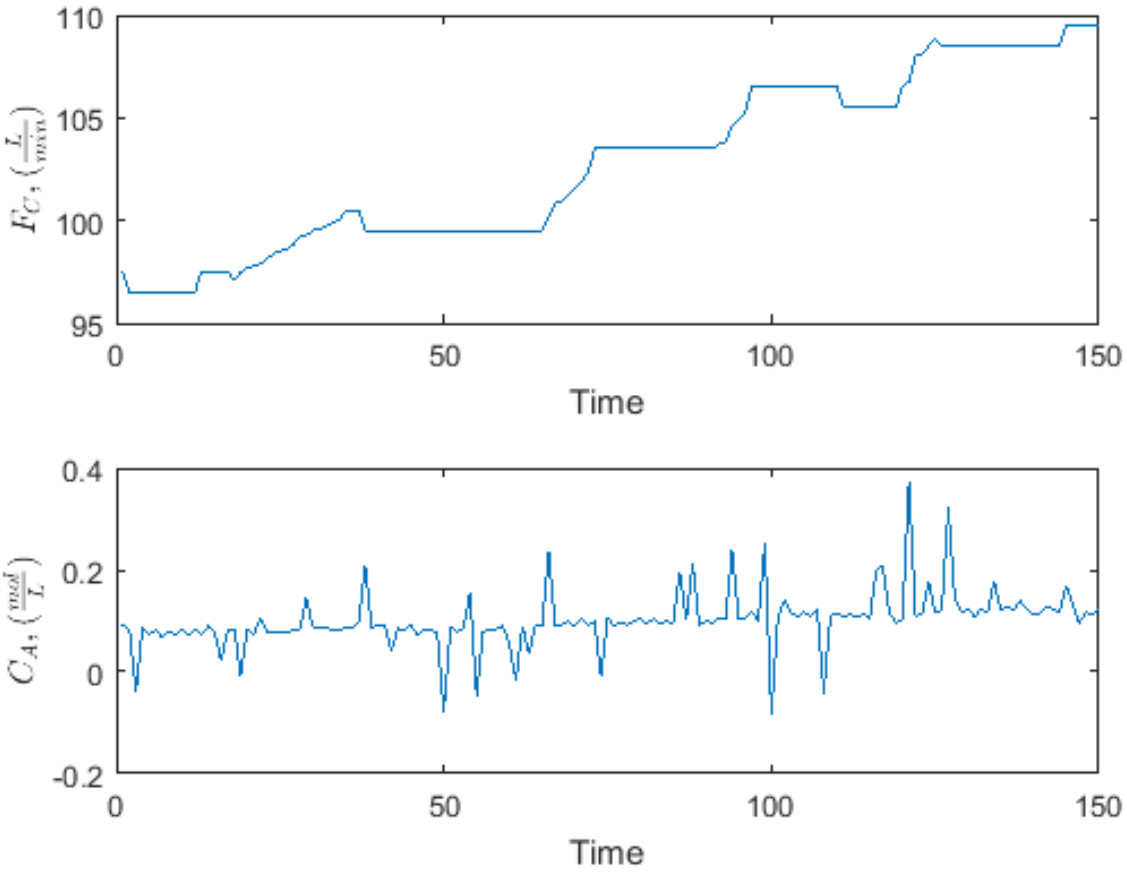

Figure 2.9: Input-output data-set used for training the CSTR model

The parameters of the model are listed in Table 2.5. This CSTR example has one input, coolant flow rate (F_C), and two outputs including: product concentration of component A (C_A), and Reactor temperature (T). However, in this simulation, we attempt to find the underlying function which maps the input to one of the outputs, i.e. product concentration of component A (C_A). The training data set used in this simulation is depicted in Fig.(2.9).

To assess the efficiency and performance of the proposed model, a mixture of two Gaussian noises is added to the output of the CSTR simulation. This mixture noise is generated using Gaussian distributions with zero mean and the variances 4×10^{-5} and 9×10^{-3} , respectively. The prediction performance of the model is presented in Fig.(2.10). Table 2.6 shows the magnitudes of MAE for the standard GPR and the

Table 2.5: Parameters of the CSTR system

Description	Notation	nominal Value
Process flow rate	F	100 $\frac{L}{min}$
CSTR volume	V	100 L
Feed Concentration of component A	C_{Af}	1 $\frac{mol}{L}$
Reaction rate constant	K_0	7.2×10^{10} min^{-1}
Activation energy	E/R	1×10^{4} K
Feed temperature	T_f	350 K
Inlet coolant temperature	T_j	350 K
Heat of reaction	ΔH	-2×10^{5} $\frac{cal}{mol}$
Liquid densitis	ρ, ρ_C	1×10^{3} $\frac{g}{L}$
Specific heats	C_p, C_{pc}	1 $\frac{cal}{g.K}$
Heat transfer term	hA	7×10^{5} $\frac{cal}{min.K}$

proposed GPR model, which clearly indicate the superiority of the proposed approach.

Table 2.6: Prediction Performance of CSTR

Method	MAE on Validation data	MAE on Training data
Standard GP	0.0088	0.0050
Robust GP with a mixture noise	**0.0027**	**0.0019**

2.7.3 Industrial Process modelling

The effectiveness of our proposed method is further investigated through an industrial modelling problem. The steam assisted gravity drainage (SAGD), a process which is used to extract heavy oil or bitumen from underground, in Northern Alberta, Canada, is considered in this section. Due to high viscosity of heavy oil or bitumen, non conventional oil recovery techniques such as SAGD improve the flow properties of bitumen by reducing the viscosity, thereby, aiding the extraction. This procedure involves the drilling of two horizontal well placed one over the other. Low pressure steam is injected into the upper well named injection well. As the steam flows upward a cone-shaped steam chamber is formed. Steam causes the bitumen to heat up, which reduces its viscosity, and enables the bitumen to flow downward into production well by the force of gravity, forming oil in water emulsion. The emulsion flow that measures water and oil mixture flow is a key variable in a SAGD process and its prediction is deemed to be critical.

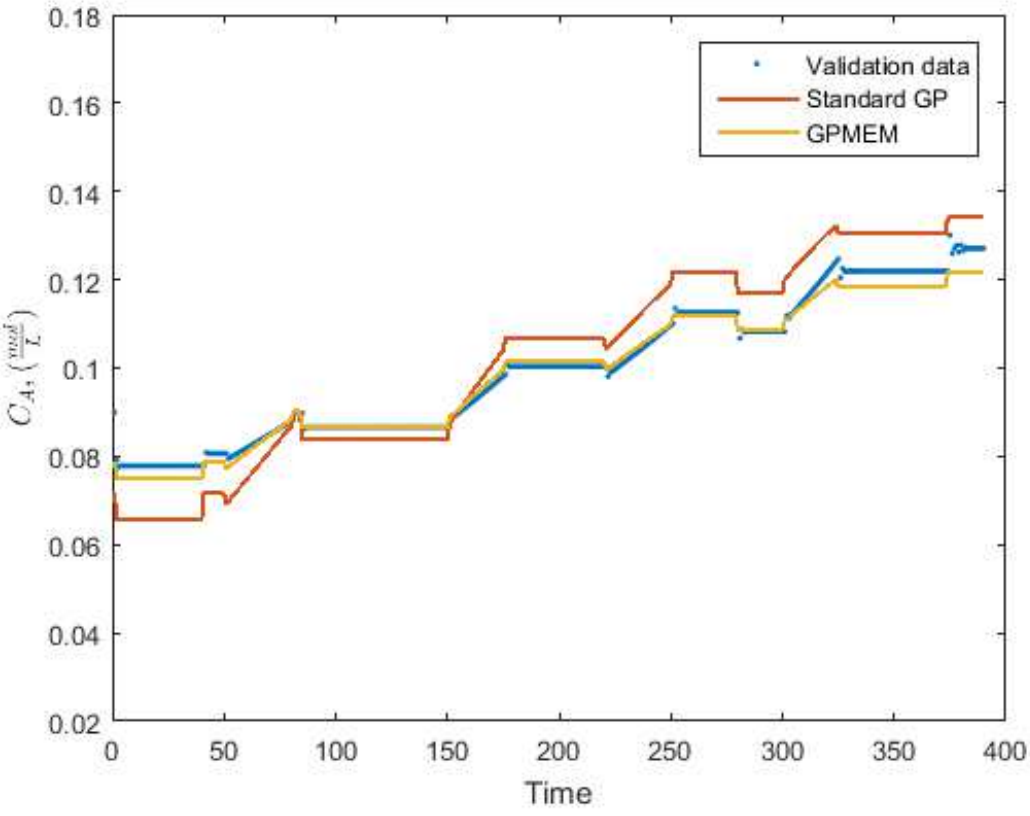

Figure 2.10: Cross-validation for CSTR data

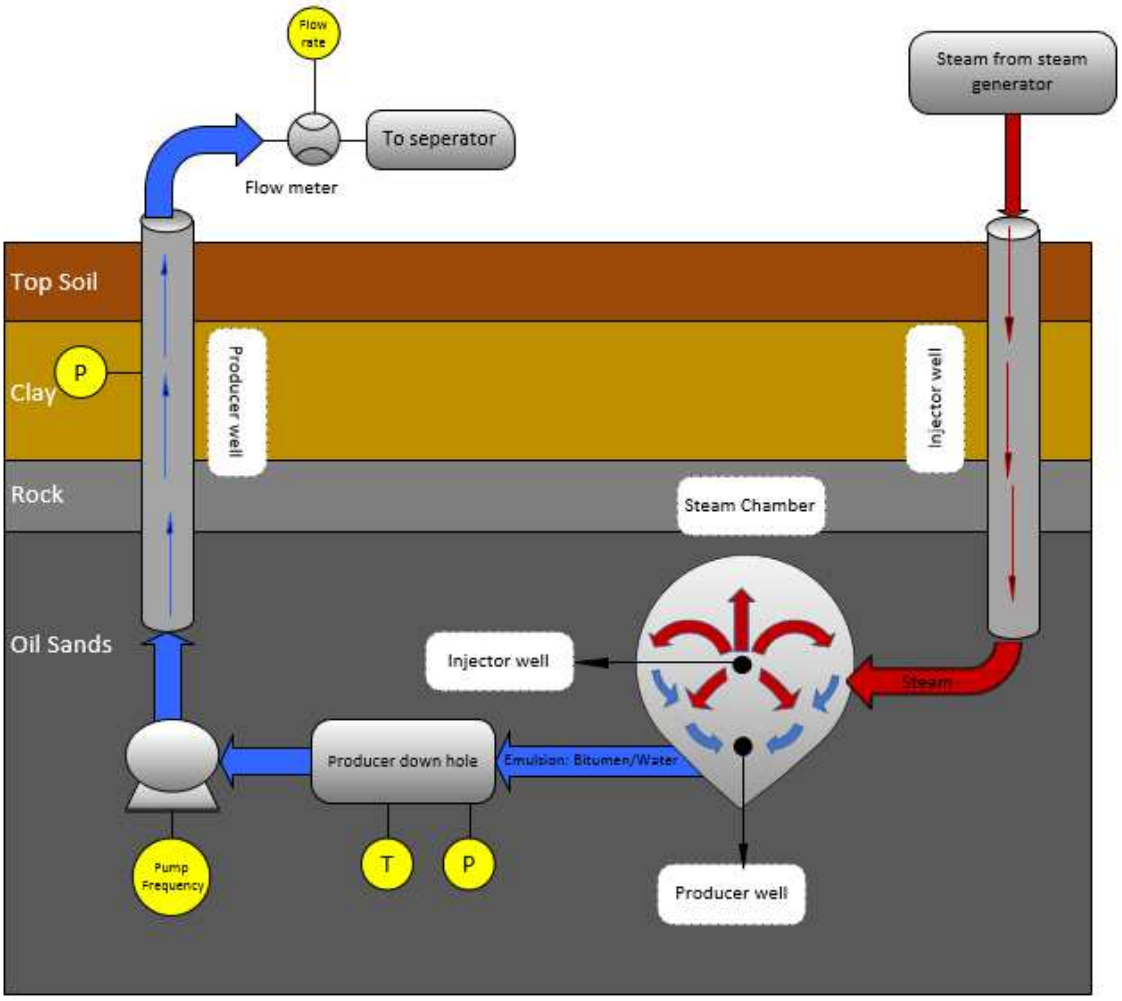

Figure 2.11: The process diagram of Emulsion flow soft sensor

We consider measurements of four most informative inputs for predicting emulsion flow (EF) rate. The measurement is averaged over 5 hours to remove unwanted peaks in process variables as part of pre-processing. Further, the data is normalized to preserve propitiatory information. The process schematic of SAGD process with relevant inputs and output is illustrated in Fig.(2.11). Standard GP and the proposed GP with the mixture likelihood is applied to model the EF rate. A statistical comparison of our proposed method with the standard GP is presented in Table 2.7. Also, Fig.(2.12) shows the cross validation results. The results presented above show the advantage of the proposed approach.

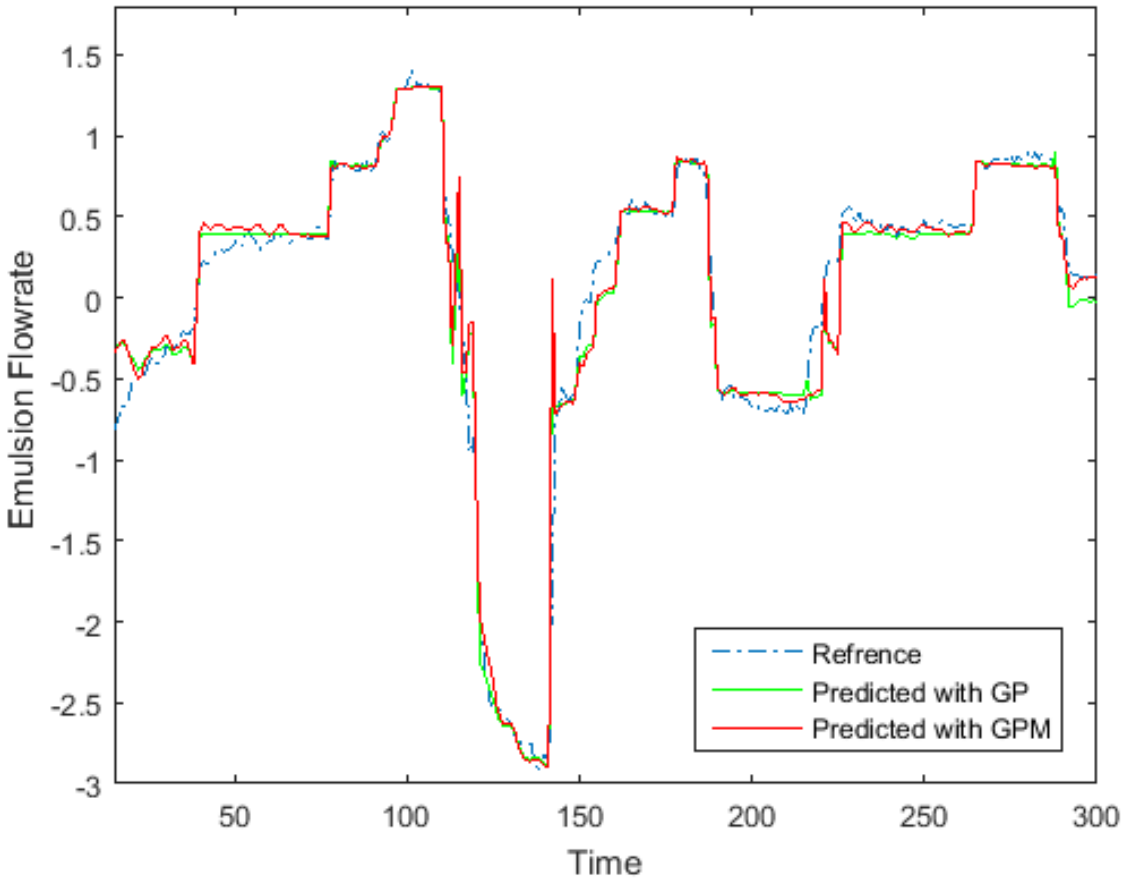

Figure 2.12: Cross-validation of the two models on SAGD Process

Table 2.7: Prediction Performance of SAGD Process

Method	MAE for validation data	MAE for training data
Standard GP	0.1260	0.0700
Robust GP with a mixture noise	**0.1221**	**0.0639**

2.8 Conclusion

In this chapter, a Robust GPR with a mixture of Gaussian likelihood has been proposed to model the processes affected by multi-modal noise. Further, we presented an approach based on EM algorithm to obtain the point estimation of the proposed model parameters. Two numerical examples and a simulated chemical process have been used to demonstrate the advantage of the proposed method. In addition, the method is applied to model industrial data from SAGD process, which further verified effectiveness of the proposed approach.

Chapter 3

Robust Gaussian Process Regression with Noisy Input using EM Algorithm

In the traditional formulation of Gaussian Process Regression (GPR), the input data is not corrupted by noise. However, this assumption is not realistic in many practical problems. A new robust GPR model is introduced in this chapter where the input and output are both corrupted by noise. To address this problem, the output noise is modelled using a mixture of two Gaussian distributions and the input noise is assumed to be an independently and identically distributed (i.i.d.) Gaussian noise. The outliers are taken into account by considering a mixture of two Gaussian distributions for the output noise. Note that this consideration renders the model to be robust against outliers. A learning scheme based on the errors-in-variables (EIV) model and the Expectation Maximization (EM) algorithm is proposed to derive a new Gaussian Process model whose kernel function is dependent on hyper-parameters in both the input and output noises. The first and second moments of the predictive distribution are obtained based on the learned hyper-parameters through the proposed algorithm. The numerical and simulation examples presented here demonstrate the practicality of the proposed method and its superiority to other existing methods.

3.1 Introduction

Nonlinear regression analysis arises in a broad class of scientific disciplines, such as engineering, statistics, biology, etc. It is performed by placing a prior on the

underlying function followed by data observation to obtain a posterior. Gaussian process regression (GPR) models are an alternative approach to traditional regression analysis, in which a multivariate Gaussian distribution is assumed as a prior for the underlying function that can be updated as data are observed to produce the posterior distribution over the function [Murphy, 2012]. Rasmussen [1999] has shown that Gaussian process (GP) is a good method for nonlinear regression and outperforms the other regression methods.

Standard GPs have been utilized successfully in regression analysis by considering two assumptions about the noises in the data; (i) the input data are noise free, and (ii) the output data are corrupted by the noise following an independent, identically distributed Gaussian distribution with constant variance [Rasmussen and Williams, 2006]. However, these assumptions might not lead to a good predictive performance in many practical regression problems, due to sampling errors, the occurrence of outliers and non-Gaussian distributed measurement errors.

Following other robust regression models, a noise likelihood with heavier tails was adopted rather than a Gaussian distribution (which violates the second assumption of Standard GP) in order to render the GP regression model robust with respect to the occurrence of outliers. Hyper-parameters for GP regression were determined by maximizing the log marginal likelihood. However, in the case of using heavy-tailed distributions, the maximization of the log marginal likelihood is intractable. Many published articles in literature have focused on learning GPR with non-Gaussian likelihood such as in Neal [1997] where the output noise is assumed to follow a t-distribution and the regression model is implemented using the Markov Chain Monte Carlo (MCMC) method. Further, Kuss [2006], Ranjan et al. [2016], Vanhatalo et al. [2009], and Jylänki et al. [2011] proposed other approximate Bayesian inferences such as: Expectation Propagation (EP), Laplace approximation, and Variational Inference to implement GPR with a student-t distribution likelihood. In Chapter 2, the GPR model was made robust to the outliers by using a mixture of two Gaussian distributions for the output noise and an EM algorithm-based approach was proposed for learning hyper-parameters.

On the other hand, error-in-variables (EIV) methods study the violation of the first assumption. In this case, both input and output are contaminated by noise.

There are two types of EIV models: (i) Classical model (ii) Berkson's model. In the classical model, the errors arise due to the measuring device and the measurements are the noise-corrupted version of the true value; while in Berkson's model the observed values are assumed to be certain and the unobserved true values vary around the certain value due to an error independent from the observed value [Carroll et al., 2006]. EIV regression models have been extensively studied in the literature. Fuller [2009] and Cheng et al. [1999] investigated the classical analysis of EIV regression models wherein maximum likelihood and least squares solution were discussed. The maximum likelihood approach for estimating the EIV model parameters was first utilized by Lindley [1947] and has been adopted by many other researchers since then. Furthermore, the total least squares method that is also used for estimating the EIV model parameters was first introduced by Golub and Van Loan [1980]. In Dellaportas and Stephens [1995], the MCMC technique was employed to analyze nonlinear EIV regression for both the Berkson and Classical models. It should be noted that the EIV models for nonlinear regression are discussed in Carroll et al. [2006].

To the best of our knowledge, there are few works on the simultaneous consideration of input and output data uncertainties in nonlinear regression. The authors in Tresp et al. [1994] trained a neural network with uncertain inputs and derived the closed form solution for a certain Gaussian basis function. Their solution separately estimated the input density with a Gaussian mixture model (GMM) and the conditional density with a feed-forward network. Also, a new robust support vector regression model with uncertain input and output for both linear and nonlinear cases, has been derived in Huang et al. [2012]. Girard and Murray-Smith [2003] focused on learning a GP model with uncertain inputs. They approximated the GP latent function f about an input using a Taylor series. Then, a new Gaussian process was defined with a corrected covariance function which accounts for the input noise. For prediction at a new random input, Girard and Murray-Smith [2003] utilized two equations for the mean and variance of the predictive distribution which has been derived in more detail in Girard et al. [2002] and Girard et al. [2003a]. McHutchon and Rasmussen [2011] also considered a GPR model with noisy input wherein some approximations were made to make the model tractable. Similar to Tresp et al. [1994], McHutchon and Rasmussen [2011] takes advantage of a local linear expansion

to transform the input noise into the output noise. They referred the input noise to output in order to reshape all the noises as output noise, leading to the output noise variance changes along with the input space (so-called heteroscedasticity). Tran et al. [2015] studied the GPR with noisy input and output for a specific problem that dealt with temporal data. In Tran et al. [2015], a variational inference-based approach was used to estimate the model, by assuming a new constraint for the problem. In the statistics literature, the case of input-dependent noise for regression models has been examined under the name of heteroscedasticity. For instance, Goldberg et al. [1998] utilized a second GP to model the noise variance, thereby employing two GPs to make predictions.

Even though there are various works on learning GPs with noisy input (Girard and Murray-Smith [2003], McHutchon and Rasmussen [2011]), the problem of robust GP with noisy input in the presence of outliers has not been considered, although practical data are often corrupted by outliers. In this chapter, a robust GPR model with noisy input is proposed. It should be noted that the formulation in (Girard and Murray-Smith [2003], McHutchon and Rasmussen [2011]) considered that both input and output measurements are corrupted by i.i.d. Gaussian noise. Here, we extend these researches to the case which is robust to the outliers using a mixture of two Gaussian distribution. Further, we develop the EM algorithm formulation to learn hyper-parameters of the proposed robust GPR model with noisy input. Parameters of a model can be estimated by maximum likelihood estimation (MLE). However, when the observed data is incomplete or contains hidden or latent variables, the marginal likelihood function can be maximized using the EM algorithm [Dempster et al., 1977]. Finally, we validate our results with synthetic data and a simulated example.

The rest of this chapter is organized as follows: the problem is described in Section 3.2. In Section 3.3 some methods for approximating the underlying process with a new GP are introduced. An EM-based approach is proposed in Section 3.4, in which the E step and the M step are presented to learn all the hyper-parameters. The predictive distribution model based on the learned hyper-parameters is also discussed. In order to verify the performance of the proposed regression model, one numerical example and one simulation example are presented in Section 3.5 and the results are discussed. Some concluding remarks are presented in Section 3.6.

3.2 Overview of the Problem

Consider a dataset $\mathcal{D}$ made up of each data vector, $\mathbf{x_i} = [x_{ip}] \in \mathbb{R}^d$ where $p = [1, 2, \ldots, d]$, with the corresponding outputs $y_i \in \mathbb{R}$:

$$\begin{aligned}\mathcal{D} &= \{(\mathbf{x_i}, y_i)|i = 1, 2, \ldots, n\} \\ \mathbf{X} &= \{\mathbf{x_i}|i = 1, 2, \ldots, n\} \\ \mathbf{y} &= \{y_i|i = 1, 2, \ldots, n\}.\end{aligned}$$

where the observed data, $\mathcal{D}$, includes $\mathbf{X}$ and $\mathbf{y}$ which are an $n \times d$ matrix and an $n \times 1$ column vector, respectively. Given this dataset, we consider the following nonlinear regression model:

$$\begin{aligned}y_i &= f(\mathbf{u_i}) + \varepsilon_{y_i} \\ \mathbf{u_i} &= \mathbf{x_i} + \boldsymbol{\varepsilon_{x_i}}\end{aligned} \tag{3.1}$$

where ε_{y_i} is a random variable representing noise in the output, $\boldsymbol{\varepsilon_{x_i}}$ is the input noise and $\mathbf{u_i} = [u_{ip}] \in \mathbb{R}^d$ is a vector of the actual input variables to the process, which are not observed. In this setting, $\mathcal{D}$ are the noise-free versions of actual inputs to the process and the noise-corrupted outputs of the process. The output noise here is assumed to be a mixture of Gaussian distributions to account for outliers along with regular noise. We further assume that the input noise for each input dimension follows an i.i.d Gaussian distribution,

$$\boldsymbol{\varepsilon_{x_i}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma_x}) \qquad \text{and} \qquad \varepsilon_{y_i} \sim \begin{cases} \mathcal{N}\left(0, \sigma_1^2\right) & w.p.\ \pi_1 \\ \mathcal{N}\left(0, \sigma_2^2\right) & w.p.\ \pi_2 \end{cases} \tag{3.2}$$

where $\boldsymbol{\Sigma_x}$ is a diagonal matrix with entries σ_x^2 on the main diagonal of the matrix, σ_1^2 denotes a high variance accounting for the outlying observations and σ_2^2 is the symbol for regular noise variance. $I = [I_1, I_2, \ldots, I_i, \ldots, I_n]$ is a vector of the model indicator in different samples as in our case $I_i \in \{1, 2\}$, representing the identity of the mixture component that has generated the output noise for any sample. π_q is used to denote the probability, $P(I_i = q); q \in [1, 2]$, that are the mixture weights. Since these mixture weights represent the probability of occurring each of outliers and regular noise, $\sum_{q=1}^{2} \pi_q = 1$ holds. We assume a GP prior on $f(.)$, which means that any finite number of the actual outputs of the process evaluated from f have a multivariate normal distribution that is entirely defined by its mean and covariance

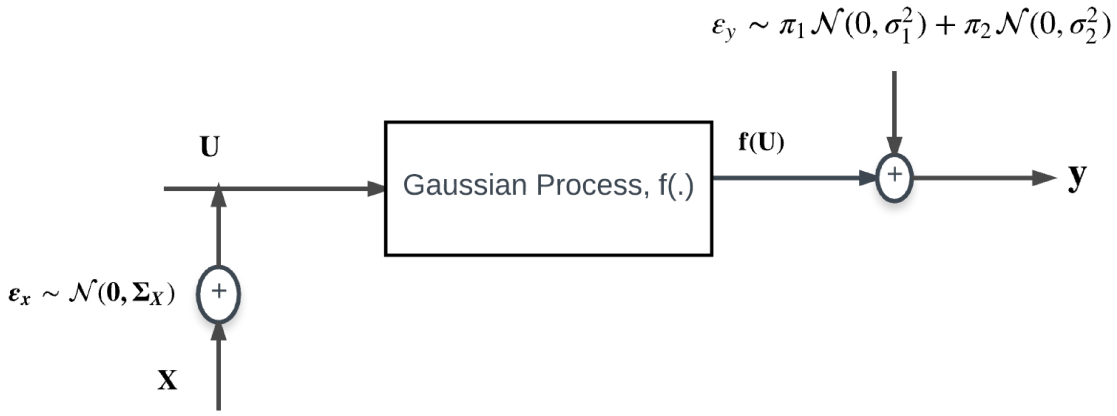

Figure 3.1: Graphical model of regression with input and output noise

function. In this case, a zero mean and squared exponential kernel is chosen for the mean and covariance functions, respectively. In this setting, the covariance function $K(U, U)$ is an $n \times n$ matrix whose entries based on the SE kernel have the form:

$$k(\mathbf{u_i}, \mathbf{u_j}) = Cov(f(\mathbf{u_i}), f(\mathbf{u_j})) = \sigma_f^2 \exp[-\sum_{p=1}^{d} \frac{(u_{ip} - u_{jp})^2}{2l_p^2}] \tag{3.3}$$

where σ_f is the signal variance, and l_p is the length scale parameter for each input space dimension p. Fig.(3.1) illustrates the GPR model with Gaussian input noise and a mixture of two Gaussian distributions as the output noise.

Let the complete set of hyper-parameters for d-dimensional input be,

$$\boldsymbol{\varphi} = [\underbrace{\ell_1, \ell_2, \ldots, \ell_d, \sigma_f}_{\phi_{GP}}, \underbrace{\sigma_x}_{\phi_{IN}}, \underbrace{\sigma_1^2, \sigma_2^2, \pi_1, \pi_2}_{\phi_{ON}}] = [\phi_{GP}, \phi_{IN}, \phi_{ON}] \tag{3.4}$$

where the GP prior, the input noise, and the output noise are parametrized by ϕ_{GP}, ϕ_{IN}, and ϕ_{ON} respectively. In learning the regression model, the optimal hyper-parameters $\hat{\boldsymbol{\varphi}}$ is estimated by maximizing the log evidence as $\hat{\boldsymbol{\varphi}} = \arg\max_{\boldsymbol{\varphi}} p(\mathbf{y}|\mathbf{X}, \boldsymbol{\varphi})$. The log evidence is reformulated using marginalization and Chain rule of probability,

$$\begin{aligned} \hat{\boldsymbol{\varphi}} &= \arg\max_{\boldsymbol{\varphi}} \iint p(\mathbf{y}, \mathbf{f}(\mathbf{U}), \mathbf{I}|\mathbf{X}, \boldsymbol{\varphi}) \; d\mathbf{f} \; d\mathbf{I} \\ &= \arg\max_{\boldsymbol{\varphi}} \iint p(\mathbf{y}|\mathbf{f}(\mathbf{U}), \mathbf{I}, \boldsymbol{\varphi}) p(\mathbf{f}(\mathbf{U})|\mathbf{X}, \boldsymbol{\varphi}) p(\mathbf{I}|\boldsymbol{\varphi}) \; d\mathbf{f}(\mathbf{U}) \; d\mathbf{I} \end{aligned} \tag{3.5}$$

The challenge resulting from the presence of the input noise is due to the nonlinear dependencies of the covariance function of the prior on the latent actual input. Therefore, the probability distribution of $P(\mathbf{f}(\mathbf{U})|\mathbf{X}, \boldsymbol{\varphi})$ will be of a non-Gaussian distribution, which makes the exact inference for Eq.(3.5) analytically intractable. Likewise, the mixture noise model also makes the log evidence non-Gaussian since the noise distribution is not log-concave. Hence, in this work, we have to resort to approximate $P(\mathbf{f}(\mathbf{U})|\mathbf{X}, \boldsymbol{\varphi})$ with a Gaussian distribution and then the Expectation-Maximization (EM) algorithm is employed to estimate the optimal parameters. After the learning part, the predictive distribution is derived based on the best-fit hyper-parameters. The detailed derivation of these steps is presented in the following sections.

3.3 Approximation of the prior with a new Gaussian process

3.3.1 Local linear expansion of the latent function about each observed input point

The prior on f has been assumed to be a GP which has the kernel form, $Cov(f(\mathbf{u_i}), f(\mathbf{u_j}))$, as Eq.(3.2). Since the actual input U is latent, the defined covariance function is a nonlinear function of the latent variable resulting in $p(f(\mathbf{U})|\mathbf{X}, \boldsymbol{\varphi})$ being analytically intractable. Then, we resort to approximate this prior with a new Gaussian process whose first and second moments are derived from the Taylor series of the latent function $f(\mathbf{u_i})$ at point $\mathbf{x_i}$ as, [McHutchon and Rasmussen, 2011]

$$f(\mathbf{u_i}) \approx f(\mathbf{x_i}) + \boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}} \tag{3.6}$$

From this assumption, the input noise is of a Gaussian distribution with zero mean $\mathbb{E}[\boldsymbol{\varepsilon}_{x_i}] = 0$. Then, the expectation calculation of this expression is obtained as follows,

$$\begin{aligned} E[f(\mathbf{u_i})] &\approx E[f(\mathbf{x_i})] + \mathbb{E}[\boldsymbol{\varepsilon}_{x_i}^T].E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] \\ E[f(\mathbf{u_i})] &\approx E[f(\mathbf{x_i})] \end{aligned} \tag{3.7}$$

It is clear from the Eq.(3.7), that the mean function of the new GP prior has the same value as the mean function of the former GP prior $\mathbf{m}(\mathbf{U}) = \mathbf{m}(\mathbf{X})$. The second

moment of the new GP, covariance matrix $Cov(f(\mathbf{u_i}), f(\mathbf{u_j}))$, for entries on the main diagonal $(i = j)$ is derived as follows,

$$Var[f(\mathbf{u_i})] \approx Var[f(\mathbf{x_i}) + \boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] \tag{3.8}$$

$$Var[f(\mathbf{u_i})] \approx Var[f(\mathbf{x_i})] + Var[\boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] + 2Cov(f(\mathbf{x_i}), \boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}) \tag{3.9}$$

$$\begin{aligned} Var[f(\mathbf{u_i})] \approx\ & Var[f(\mathbf{x_i})] + E[\boldsymbol{\varepsilon}_{x_i}^T]^T Var[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] E[\boldsymbol{\varepsilon}_{x_i}^T] \\ & + E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]^T Var[\boldsymbol{\varepsilon}_{x_i}^T] E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] + Var[\boldsymbol{\varepsilon}_{x_i}^T] Var[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] \\ & + 2Cov(f(\mathbf{x_i}), \boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}) \end{aligned} \tag{3.10}$$

$$\begin{aligned} Var[f(\mathbf{u_i})] \approx\ & Var[f(\mathbf{x_i})] + E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]^T \boldsymbol{\Sigma}_x E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] \\ & + Tr(\boldsymbol{\Sigma}_x . Var[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]) \end{aligned} \tag{3.11}$$

$$\begin{aligned} Cov[f(\mathbf{u_i}), f(\mathbf{u_i})] \approx\ & Cov[f(\mathbf{x_i}), f(\mathbf{x_i})] + E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]^T \boldsymbol{\Sigma}_x E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] \\ & + Tr(\boldsymbol{\Sigma}_x . Var[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]) \end{aligned} \tag{3.12}$$

$$\begin{aligned} k(\mathbf{u_i}, \mathbf{u_i}) \approx\ & k(\mathbf{x_i}, \mathbf{x_i}) + E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]^T \boldsymbol{\Sigma}_x E[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}] \\ & + Tr(\boldsymbol{\Sigma}_x . Var[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]) \end{aligned} \tag{3.13}$$

Since the input noise is modelled independently for different samples, all the terms except the first term in Eq.(3.15) are evaluated to be zero. The covariance matrix off-diagonal elements are obtained as,

$$\begin{aligned} Cov\Big[f(\mathbf{u_i}),\ f(\mathbf{u_j})\Big] &\approx Cov\Big[f(\mathbf{x_i}) + \boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}},\ f(\mathbf{x_j}) + \boldsymbol{\varepsilon}_{x_j}^T \frac{\partial f(\mathbf{u_j})}{\partial \mathbf{u_j}}\Big|_{\mathbf{u_j}=\mathbf{x_j}}\Big] \\ &\approx Cov\Big[f(\mathbf{x_i}),\ f(\mathbf{x_j})\Big] + Cov\Big[f(\mathbf{x_i}),\ \boldsymbol{\varepsilon}_{x_j}^T \frac{\partial f(\mathbf{u_j})}{\partial \mathbf{u_j}}\Big|_{\mathbf{u_j}=\mathbf{x_j}}\Big] \\ &\quad + Cov\Big[\boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}},\ f(\mathbf{x_j})\Big] \end{aligned} \tag{3.14}$$

$$+ Cov\left[\boldsymbol{\varepsilon}_{x_i}^T \frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}},\ \boldsymbol{\varepsilon}_{x_j}^T \frac{\partial f(\mathbf{u_j})}{\partial \mathbf{u_j}}\Big|_{\mathbf{u_j}=\mathbf{x_j}}\right] \tag{3.15}$$

$$\approx Cov\Big[f(\mathbf{x_i}),\ f(\mathbf{x_j})\Big] = k(\mathbf{x_i}, \mathbf{x_j}) \tag{3.16}$$

Thus, the probability of $p(\mathbf{f}(U)|X, \boldsymbol{\varphi})$ is approximated by a new GP given as follows,

$$p(\mathbf{f}(U)|X, \boldsymbol{\varphi}) \sim \mathcal{N}\Big(\mathbf{0}, \mathbf{S}(\mathbf{X}, \mathbf{X})\Big) \tag{3.17}$$

where all the elements of the covariance matrix for the new GP prior are approximated as follows,

$$s(\mathbf{x_i}, \mathbf{x_j}) \approx k(\mathbf{x_i}, \mathbf{x_j}) + cc(\mathbf{x_i}, \mathbf{x_j}) \tag{3.18}$$

In this approximation method, $cc(\mathbf{x_i}, \mathbf{x_j})$ is derived using Eq.(3.13) and Eq.(3.16) as given below,

$$\begin{cases} cc(\mathbf{x_i}, \mathbf{x_j}) = \Omega_i & i = j \\ cc(\mathbf{x_i}, \mathbf{x_j}) = 0 & i \neq j \end{cases} \tag{3.19}$$

where $\Omega_i = \mathbb{E}\left[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}\right]^T . \boldsymbol{\Sigma}_x . \mathbb{E}\left[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}\right] + Tr\left(\boldsymbol{\Sigma}_x . \mathbb{V}[\frac{\partial f(\mathbf{u_i})}{\partial \mathbf{u_i}}\Big|_{\mathbf{u_i}=\mathbf{x_i}}]\right)$ is the corrected covariance term obtained using Eq.(3.13). We need to make another approximation here since the problem cannot be solved analytically. McHutchon and Rasmussen [2011] approximated the expectation and variance of the derivative of GP with the derivative of the GP posterior mean and variance. This method for approximating the prior on $\mathbf{f}$ is deployed in EM derivations, as well as another approximation method proposed in the following section.

3.3.2 Expectation of Taylor series expansion for the Covariance function

In this method, the Taylor series expansion for the covariance matrix $k(\mathbf{u_i}, \mathbf{u_j}) = k(\mathbf{x_i} + \boldsymbol{\varepsilon}_{x_i}, \mathbf{x}_j + \boldsymbol{\varepsilon}_{x_j})$ is used. Since the expected value of the covariance matrix is directly intractable, the Taylor series up to order 4 about the point $(\mathbf{u_i}, \mathbf{u_j})$ is written

as follows,

$$\tilde{k}(\mathbf{u_i},\mathbf{u_j}) = k(\mathbf{x_i},\mathbf{x_j}) + \boldsymbol{\varepsilon}_{\boldsymbol{x_i}}^{\boldsymbol{T}}\frac{\partial k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}} + \boldsymbol{\varepsilon}_{\boldsymbol{x_j}}^{\boldsymbol{T}}\frac{\partial k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_j}} + \frac{1}{2}\boldsymbol{\varepsilon}_{\boldsymbol{x_i}}^{\boldsymbol{T}}\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^2}\boldsymbol{\varepsilon}_{\boldsymbol{x_i}} + \frac{1}{2}\boldsymbol{\varepsilon}_{\boldsymbol{x_j}}^{\boldsymbol{T}}\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_j}^2}\boldsymbol{\varepsilon}_{\boldsymbol{x_j}} + \boldsymbol{\varepsilon}_{\boldsymbol{x_i}}^{\boldsymbol{T}}\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}\partial \mathbf{x_j}}\boldsymbol{\varepsilon}_{\boldsymbol{x_j}} + \dots\Big] \tag{3.20}$$

The expectation of the approximated covariance matrix via Taylor series is derived as follows,

$$E(\tilde{k}(\mathbf{u_i},\mathbf{u_j})) = k(\mathbf{x_i},\mathbf{x_j}) + \frac{1}{2}Tr(\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^2}\boldsymbol{\Sigma}_{\boldsymbol{x}}) + \frac{1}{2}Tr(\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_j}^2}\boldsymbol{\Sigma}_{\boldsymbol{x}}) + \frac{\sigma_x^4}{4!}\Big[3Tr(\frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^4}) + 3Tr(\frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_j}^4}) + 6Tr(\frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^2\partial \mathbf{x_j}^2})\Big] \tag{3.21}$$

Once the first and second derivatives of the kernel function are determined, the following relationship can be obtained, $\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^2} = \frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_j}^2}$ and for fourth derivative we can see this relationship $\frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^4} = \frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_j}^4} = \frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^2\partial \mathbf{x_j}^2}$ holds. By deploying these relationships and the second and fourth derivatives of kernel function into Eq.(3.21), the final form of the approximation for kernel function is derived as,

$$\begin{aligned} E(\tilde{k}(\mathbf{u_i},\mathbf{u_j})) &= k(\mathbf{x_i},\mathbf{x_j}) + \sigma_x^2 Tr(\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^2}) + \sigma_x^4 Tr(\frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial \mathbf{x_i}^4}) \\ &= k(\mathbf{x_i},\mathbf{x_j}) + \sigma_x^2[\frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial x_{i1}^2} + \dots + \frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial x_{id}^2}] + \sigma_x^4[\frac{\partial^4 k(\mathbf{x_i},\mathbf{x_j})}{\partial x_{i1}^4} + \dots + \frac{\partial^2 k(\mathbf{x_i},\mathbf{x_j})}{\partial x_{id}^4}] \\ &= k(\mathbf{x_i},\mathbf{x_j}) + \underbrace{k(\mathbf{x_i},\mathbf{x_j})[\sigma_x^2\sum_{p=1}^{d}(\frac{(x_{ip}-x_{jp})^2}{l_p^4} - \frac{1}{l_p^2}) + \sigma_x^4\sum_{p=1}^{d}(\frac{(x_{ip}-x_{jp})^4}{l_p^8} - \frac{6(x_{ip}-x_{jp})^2}{l_p^6} + \frac{3}{l_p^4})]}_{cc(x_i,x_j)} \end{aligned} \tag{3.22}$$

In this case, the new GP prior has also the form as Eq.(3.17), wherein the covariance matrix term entries can be presented as,

$$s(\mathbf{x_i},\mathbf{x_j}) \approx k(\mathbf{x_i},\mathbf{x_j}) + cc(x_i,x_j) \tag{3.23}$$

Fig.(3.2), shows a schematic of the proposed regression model with input and output noise which introduces a new pathway to inference the input noise parameters, as well as GP and output noise parameters for the two approximation methods.

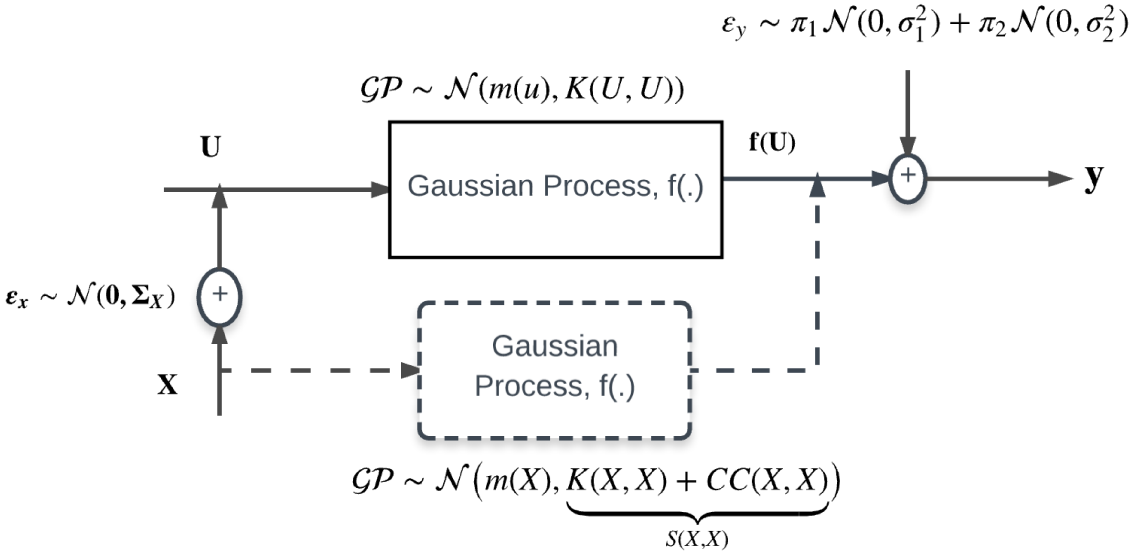

Figure 3.2: Graphical model of regression from another pathway

3.4 Robust GPR model with noisy input using the EM algorithm

3.4.1 E-Step derivation:

Now we proceed to solve the problem using the new GPs obtained from Section 3.3. Another challenge in deriving the Eq.(3.5) is that $f(U)$ and I are hidden, and need to be estimated. These variables are replaced with their conditional expectation given the observed data. Thus, the posterior of $\mathbf{f}(\mathbf{U})$ and $\mathbf{I}$ need to be derived. The posterior $\mathbf{f}(\mathbf{U})$ which is based on Bayes' rule is given by,

$$p(\mathbf{f}(\mathbf{U})|\mathcal{D}, \mathbf{I}, \boldsymbol{\varphi}^{(t)}) = \frac{p(\mathbf{f}(\mathbf{U})|X, \boldsymbol{\varphi}^{(t)}) p(\mathbf{y}|\mathbf{f}(\mathbf{U}), \mathbf{I}, \boldsymbol{\varphi}^{(t)})}{p(\mathbf{y}|X, \mathbf{I}, \boldsymbol{\varphi}^{(t)})} \tag{3.24}$$

where the prior over $\mathbf{f}$ given the new GP prior obtained from Section 3.3, is $p(\mathbf{f}(\mathbf{U})|X, \boldsymbol{\varphi}^{(t)}) \sim \mathcal{N}\Big(\mathbf{0}, \mathbf{S}(\mathbf{X}, \mathbf{X})\Big)$. Also, the mixture of two Gaussian noises distribution given the noise mode identity in the multivariate vector form is $p(\mathbf{y}|\mathbf{f}(\mathbf{U}), \mathbf{I}, \boldsymbol{\varphi}) \sim \mathcal{N}\Big(\mathbf{0}, (diag(\sigma_{\mathbf{I}}))^2\Big)$. Since the denominator is independent of $\mathbf{f}$, the posterior can be approximated as,

$$\begin{aligned} p(\mathbf{f}(\mathbf{U})|\mathcal{D}, \mathbf{I}, \boldsymbol{\varphi}) \approx q(\mathbf{f}(\mathbf{U})) &\propto p(\mathbf{f}(\mathbf{U})|X, \boldsymbol{\varphi}) p(\mathbf{y}|\mathbf{f}(\mathbf{U}), \mathbf{I}, \boldsymbol{\varphi}) \\ &\propto \mathcal{N}(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{V}}) \end{aligned} \tag{3.25}$$

where $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{V}}\Big((diag(\sigma_{\mathbf{I}}))^{-2}\mathbf{y}\Big)$ and $\boldsymbol{\mathcal{V}} = \Big(S(X,X)^{-1}+(diag(\sigma_{\mathbf{I}}))^{-2}\Big)^{-1}$. The posterior of $\mathbf{I}$ is also written using Bayes' rule as follows,

$$\begin{aligned} p(\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)}) &= \frac{p(\mathbf{y}|X,\mathbf{I},\boldsymbol{\varphi}^{(t)})p(\mathbf{I}|\boldsymbol{\varphi}^{(t)})}{p(\mathbf{y}|X,\boldsymbol{\varphi}^{(t)})} \\ &= \frac{p(\mathbf{y}|X,\mathbf{I},\boldsymbol{\varphi}^{(t)})p(\mathbf{I}|\boldsymbol{\varphi}^{(t)})}{\sum_{I_1=1}^{2}\sum_{I_2=1}^{2}\cdots\sum_{I_n=1}^{2} p(\mathbf{y}|X,\mathbf{I},\boldsymbol{\varphi}^{(t)})p(\mathbf{I}|\boldsymbol{\varphi}^{(t)})} \end{aligned} \tag{3.26}$$

where the vector $\mathbf{I}$ has entries I_i, which is the noise mode identity for any sample. For mixture of two Gaussian noises, each entry can take two values which are 1 or 2. Then, there exist 2^n ordered arrangements of the elements of a set $\mathbf{I}$ with length n. In order to make the problem analytically tractable, this posterior must be approximated. We assume elements of the latent variable $\mathbf{I}$ are mutually independent and each is governed by a distinct density.

$$P(I_i = q|\mathcal{D}_i,\varphi^{(t)}) = \frac{\frac{1}{\sqrt{2\pi\sigma_q^{(t)^2}}}\exp[-\frac{(y_i-\tilde{f}_i^{(t)})^2}{2\sigma_q^{(t)^2}}]\ \alpha_q^{(t)}}{\sum_{q=1}^{2}\frac{1}{\sqrt{2\pi\sigma_q^{(t)^2}}}\exp[-\frac{(y_i-\tilde{f}_i^{(t)})^2}{2\sigma_q^{(t)^2}}]\ \alpha_q^{(t)}} \triangleq \lambda_{iq}^{(t)} \tag{3.27}$$

where during deriving this approximated posterior for I_i, we further assume that the approximate value of $\tilde{f}_i$ can be replaced with the mean of the posterior of $f(\mathbf{u_i})$, which is $\mathcal{A}_i^{(t-1)}$.

Given the posterior distribution of the hidden variables was obtained, Eq.(3.5) is revisited, and a lower bound for the marginal likelihood is constructed as below,

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X},\boldsymbol{\varphi}) &= \log\sum_I \int_{\mathbf{f}(\mathbf{U})} p(\mathbf{y},\mathbf{f}(\mathbf{U}),\mathbf{I}|X,\boldsymbol{\varphi})\ d\mathbf{f}(\mathbf{U}) \\ &= \log\sum_I \int_{\mathbf{f}(\mathbf{U})} p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})\frac{p(\mathbf{y},\mathbf{f}(\mathbf{U}),\mathbf{I}|X,\boldsymbol{\varphi})}{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\ d\mathbf{f}(\mathbf{U}) \\ &= \log\mathbb{E}_{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\left[\frac{p(\mathbf{y},\mathbf{f}(\mathbf{U}),\mathbf{I}|X,\boldsymbol{\varphi})}{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\right] \end{aligned}$$

Based on Jensen's inequality,

$$\begin{aligned} &\geqslant \mathbb{E}_{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\left[\log\frac{p(\mathbf{y},\mathbf{f}(\mathbf{U}),\mathbf{I}|X,\boldsymbol{\varphi})}{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\right] \\ &= \underbrace{\mathbb{E}_{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\Big[\log p(\mathbf{y},\mathbf{f}(\mathbf{U}),\mathbf{I}|X,\boldsymbol{\varphi})\Big]}_{\mathcal{F}(\boldsymbol{\varphi};\boldsymbol{\varphi}^{(t)})} \underbrace{-\mathbb{E}_{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\Big[\log p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)}))\Big]}_{\geqslant 0} \end{aligned} \tag{3.28}$$

Thus, $\mathcal{F}(\boldsymbol{\varphi}; \boldsymbol{\varphi}^{(t)})$ is a lower bound of marginal likelihood which is obtained as follows:

$$\mathcal{F}(\boldsymbol{\varphi}; \boldsymbol{\varphi}^{(t)}) = \mathbb{E}_{p(\mathbf{f},\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\Big[\log\Big[p(\mathbf{y}|\mathbf{f}(\mathbf{U}),\mathbf{I},\boldsymbol{\varphi})p(\mathbf{f}(\mathbf{U})|\mathbf{X},\boldsymbol{\varphi})p(\mathbf{I}|\boldsymbol{\varphi})\Big]\Big]$$

First, $\mathbf{f}(\mathbf{U})$ is estimated by its conditional expectation as derived below,

$$\begin{aligned}
&= -\frac{n}{2}\log 2\pi - \mathbb{E}_{p(\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\Big\{\log|diag(\sigma_{\mathbf{I}})|\Big\} \\
&- \frac{1}{2}\mathbb{E}_{p(\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\Big\{Tr\Big(\big(diag(\sigma_{\mathbf{I}})\big)^{-2}\boldsymbol{\mathcal{V}}^{(\mathbf{t})}\Big)\Big\} \\
&- \frac{1}{2}\mathbb{E}_{p(\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\Big\{(\mathbf{y}-\boldsymbol{\mathcal{A}}^{(\mathbf{t})})^T\big(diag(\sigma_{\mathbf{I}})\big)^{-2}(\mathbf{y}-\boldsymbol{\mathcal{A}}^{(\mathbf{t})})\Big\} \\
&- \frac{1}{2}\log|2\pi(\mathbf{S}(\mathbf{X},\mathbf{X}))| - \frac{Tr\Big((\mathbf{S}(\mathbf{X},\mathbf{X}))^{-1}\boldsymbol{\mathcal{V}}^{(\mathbf{t})}\Big)}{2} \\
&- \frac{\boldsymbol{\mathcal{A}}^{(\mathbf{t})^T}(\mathbf{S}(\mathbf{X},\mathbf{X}))^{-1}\boldsymbol{\mathcal{A}}^{(\mathbf{t})}}{2} + \mathbb{E}_{p(\mathbf{I}|\mathcal{D},\boldsymbol{\varphi}^{(t)})}\Big\{\log(P(\mathbf{I}|\theta_n))\Big\}
\end{aligned} \tag{3.29}$$

Second, using the conditional expectation of I_i, the hidden I_i is approximated. The final form of the lower bound after taking the expectation is given below,

$$\begin{aligned}
\mathcal{F}(\boldsymbol{\varphi}; \boldsymbol{\varphi}^{(t)}) &= -\frac{n}{2}\log 2\pi - \sum_{q=1}^{2}\sum_{i=1}^{n}\lambda_{iq}^{(t)}\log\sigma_{I_i=q} - \frac{1}{2}\sum_{q=1}^{2}\sum_{i=1}^{n}\lambda_{iq}^{(t)}\left(\frac{\mathcal{V}_{ii}^{(t)}}{\sigma_{I_i=q}^2}\right) \\
&- \frac{1}{2}\sum_{q=1}^{2}\sum_{i=1}^{n}\lambda_{iq}^{(t)}\left(\frac{(y_i-\mathcal{A}_i^{(t)})^2}{\sigma_{I_i=q}^2}\right) \\
&- \frac{1}{2}\log|2\pi(\mathbf{S}(\mathbf{X},\mathbf{X}))| - \frac{Tr\Big((\mathbf{S}(\mathbf{X},\mathbf{X}))^{-1}\boldsymbol{\mathcal{V}}^{(\mathbf{t})}\Big)}{2} \\
&- \frac{\boldsymbol{\mathcal{A}}^{(\mathbf{t})^T}(\mathbf{S}(\mathbf{X},\mathbf{X}))^{-1}\boldsymbol{\mathcal{A}}^{(\mathbf{t})}}{2} + \sum_{q=1}^{2}\sum_{i=1}^{n}\lambda_{iq}^{(t)}\log\alpha_{I_i=q}
\end{aligned} \tag{3.30}$$

3.4.2 M-Step derivation

Let ϕ be the set of all parameters except output noise parameters, $[\phi_{GP}, \phi_{IN}]$. Thus, the derivative of the lower bound presented in Eq.(3.30) w.r.t ϕ can be derived as:

$$\begin{aligned}
\frac{\partial\mathcal{F}(\boldsymbol{\varphi}; \boldsymbol{\varphi}^{(t)})}{\partial\phi} &= -\frac{1}{2}Tr(\mathbf{S}^{-\mathbf{1}}\frac{\partial\mathbf{S}}{\partial\phi}) + \frac{1}{2}(\boldsymbol{\mathcal{A}}^{(t)})^T\mathbf{S}^{-\mathbf{1}}\frac{\partial\mathbf{S}}{\partial\phi}\mathbf{S}^{-\mathbf{1}}(\boldsymbol{\mathcal{A}}^{(t)}) \\
&+ \frac{1}{2}Tr(\mathbf{S}^{-\mathbf{1}}\frac{\partial\mathbf{S}}{\partial\phi}\mathbf{S}^{-1}\boldsymbol{\mathcal{V}}^{(\mathbf{t})})
\end{aligned} \tag{3.31}$$

Update equations using the local linear approximation method

This section includes the update equations for the case with the covariance matrix entries as derived in Eq.(3.18). Since the hyper-parameters of GP introduced in Section 3.2, ϕ_{GP}, include $[\underbrace{\ell_1, \ell_2, \ldots, \ell_d}_{\mathbf{\Lambda}}, \sigma_f]$, the partial derivative of the covariance matrix obtained from Eq. (3.18) with respect to GP hyper-parameters would be $\frac{\partial \mathbf{S}}{\partial \boldsymbol{\phi}_{\boldsymbol{GP}}} = \mathbf{G} \in \mathbb{R}^{n \times n \times (d+1)}$ that is derived as follows,

$$\mathbf{G_{1:n,1:n,d+1}} = \nabla_{\log \sigma_f} \mathbf{S} = \frac{\partial [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \mathbf{CC}(\mathbf{X}, \mathbf{X})]}{\partial \log \sigma_f} = 2\mathbf{K}(\mathbf{X}, \mathbf{X}) \tag{3.32}$$

where $\mathbf{G_{1:n,1:n,d+1}} = [g_{i,j,(d+1)}]$, $i = 1, 2, \ldots, n; j = 1, 2, \ldots, n$ is an $n \times n$ matrix with the elements are given below,

$$g_{i,j,(d+1)} = 2k(\mathbf{x_i}, \mathbf{x_j}) \tag{3.33}$$

The partial derivative of the covariance matrix with respect to the length scale parameters is derived as below,

$$\mathbf{G_{1:n,1:n,1:d}} = \nabla_{\log \mathbf{\Lambda}} \mathbf{S} = \frac{\partial [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \mathbf{CC}(\mathbf{X}, \mathbf{X})]}{\partial \log \mathbf{\Lambda}} \tag{3.34}$$

where $\mathbf{G_{1:n,1:n,1:d}} = [g_{i,j,k}]$, $i = 1, 2, \ldots, n; j = 1, 2, \ldots, n; k = 1, 2, \ldots, d$ is a 3D matrix $n \times n \times d$ with the following elements,

$$g_{i,j,k} = \frac{1}{l_k^2}(x_{ik} - x_{jk})^2 k(\mathbf{x_i}, \mathbf{x_j}) \tag{3.35}$$

The partial derivative of the covariance matrix with respect to the input noise parameter would be $\frac{\partial \mathbf{S}}{\partial \boldsymbol{\phi}_{\boldsymbol{IN}}} = \mathbf{H} \in \mathbb{R}^{n \times d}$ which is derived as follows,

$$\mathbf{H_{1:n,1:d}} = \nabla_{\log \sigma_x} \mathbf{S} = \frac{\partial [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \mathbf{CC}(\mathbf{X}, \mathbf{X})]}{\partial \log \sigma_x} \tag{3.36}$$

where $\mathbf{H_{1:n,1:d}} = [h_{i,k}]$, $i = 1, 2, \ldots, n; k = 1, 2, \ldots, d$ is an $n \times d$ matrix. Its elements is derived as follows,

$$h_{i,k} = 2\sigma_x^2 \left(\left[\frac{\partial \mathbb{E}[f(u_{ik})]}{\partial u_{ik}} \Big|_{u_{ik}=x_{ik}} \right] \left[\frac{\partial \mathbb{E}[f(u_{ik})]}{\partial u_{ik}} \Big|_{u_{ik}=x_{ik}} \right]^T + \left[\frac{\partial \mathbb{V}[f(u_{ik})]}{\partial u_{ik}} \Big|_{u_{ik}=x_{ik}} \right]^T \right) \tag{3.37}$$

Substituting Eqs.(3.32),(3.34), and (3.36) respectively, into Eq.(3.31) and solving using the gradient descent method, the optimal values of $\phi = [\theta_{GP}, \theta_{IN}]$ maximizing the $\mathcal{F}$ function is obtained. The closed form of the update equations for the output noise parameters is obtained by setting the derivative of the $\mathcal{F}$ function with respect to the output parameters to zero. After a little algebra, the update equations for the noise components variance, $\sigma_q^2 \in [\sigma_1^2, \sigma_2^2]$ are obtained. For the sake of brevity, we omit the details of derivation. Deriving the update equation for α_q results in a constrained optimization problem with an additional constraint: $\sum_{q=1}^{2} \pi_q = 1$. The Lagrangian multiplier is utilized to solve this constrained optimization problem. The update equations for σ_q and α_q are given below,

$$\frac{\partial \mathcal{F}(\boldsymbol{\varphi}; \boldsymbol{\varphi}^{(t)})}{\partial \sigma_q^2} = 0 \Rightarrow {\sigma_q^2}^{(t+1)} = \frac{\sum_{i=1}^{n} \lambda_{iq}^{(t)} (\mathcal{V}_{ii}^{(t)} + (y_i - \mathcal{A}_i^{(t)})^2)}{\sum_{i=1}^{n} \lambda_{iq}^{(t)}} \tag{3.38}$$

$${\alpha_q}^{(t+1)} = \frac{\sum_{i=1}^{n} \lambda_{iq}^{(t)}}{n} \tag{3.39}$$

Since the update expressions for all the parameters, have been derived, the E and M steps are iteratively solved until the converged parameters are obtained.

Update equations using the expectation of the Taylor series of covariance function:

In this section, the update equations are derived based on the case with the covariance matrix entries given in Eq.(3.23). The derivative of the lower bound is the same as the previous case in Eq.(3.31). The derivative of the covariance matrix obtained from Eq.(3.23) w.r.t all the GP parameters and input noise standard deviation is derived as,

$$\mathbf{G_{1:n,1:n,d+1}} = \nabla_{\log \sigma_f} \mathbf{S} = 2\Big[\mathbf{S}(\mathbf{X}, \mathbf{X})\Big] \tag{3.40}$$

where $\mathbf{G_{1:n,1:n,1:d}} = \nabla_{\log \mathbf{\Lambda}} \mathbf{S} = [g_{i,j,k}]$, $i = 1, 2, \ldots, n; j = 1, 2, \ldots, n; k = 1, 2, \ldots, d$ is a 3D matrix $n \times n \times d$ with elements,

$$g_{i,j,k} = \Big[k(\mathbf{x_i} - \mathbf{x_j}) + cc(\mathbf{x_i} - \mathbf{x_j})\Big] \frac{(x_{ik} - x_{jk})^2}{l_k^2} + k(\mathbf{x_i} - \mathbf{x_j}) \Bigg(\sigma_x^2 [-4 \frac{(x_{ik} - x_{jk})^2}{l_k^4} + \frac{2}{l_k^2}] + \sigma_x^4 [-8 \frac{(x_{ik} - x_{jk})^4}{l_k^8} + 36 \frac{(x_{ik} - x_{jk})^2}{l_k^6} - \frac{12}{l_k^4}] \Bigg) \tag{3.41}$$

The partial derivative of the covariance matrix with respect to the input noise parameter would be $\frac{\partial \mathbf{S}}{\partial \boldsymbol{\phi}_{IN}} = \nabla_{\log \sigma_x} \mathbf{S} = \mathbf{H} \in \mathbb{R}^{n \times n}$ where $\mathbf{H_{1:n,1:n}} = [h_{i,j}]$, $i = 1, 2, \ldots, n; j = 1, 2, \ldots, n$ is a matrix $n \times n$ with elements as follows,

$$h_{i,j} = k(\mathbf{x_i}, \mathbf{x_j}) [2\sigma_x^2 \sum_{p=1}^{d} (\frac{(x_{ip} - x_{jp})^2}{l_p^4} - \frac{1}{l_p^2}) + 4\sigma_x^4 \sum_{p=1}^{d} (\frac{(x_{ip} - x_{jp})^4}{l_k^8} - \frac{6(x_{ip} - x_{jp})^2}{l_p^6} + \frac{3}{l_p^4})] \tag{3.42}$$

The derivatives of the lower bound w.r.t output noise parameters are the same as the former case.

3.4.3 The predictive distribution

Using the linear expansion of latent function

Once the set of all parameters is estimated from update equations for the first case, the predictive distributions can be constructed based on the fitted parameters. In order to compute the predictive distribution model, we need to first derive the joint distribution of the predicted value at the new points and observed value given the inputs and the parameters estimated from M-step as,

$$P(\mathbf{y}, \mathbf{f}_* | \mathbf{X}, \mathbf{X}_*, \Phi) \sim \mathcal{N}(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{S}(\mathbf{X}, \mathbf{X}) + \Big(diag(\sigma_{\mathbf{I}})\Big)^2 & \mathbf{K}(\mathbf{X}, \mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*, \mathbf{X}) & \mathbf{S}(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}) \tag{3.43}$$

where $\mathbf{S}(\mathbf{X}_*, \mathbf{X}_*)$ is the approximated covaraince matrix of $\mathbf{f}_*$, whose entries are derived using Eq.(3.18), and $\mathbf{K}(\mathbf{X}, \mathbf{X}_*)$ is the covariance between $\mathbf{f}_*$ and $\mathbf{y}$. By the probability rules for conditioning Gaussian, the predictive distribution can be derived

as,

$$
\begin{aligned}
&P(\mathbf{f}_*|\mathbf{y},\mathbf{X},\mathbf{X}_*,\Phi) \sim \mathcal{N}(\boldsymbol{n_*},\mathbf{V}_*) \\
&\quad where \qquad \boldsymbol{n_*} = \mathbf{K}(\mathbf{X}_*,\mathbf{X})\Big(\mathbf{S}(\mathbf{X},\mathbf{X}) + \Big(\mathbf{diag}(\sigma_\mathbf{I})\Big)^\mathbf{2}\Big)^{-1}\mathbf{y}, \\
&\qquad\qquad\quad \mathbf{V}_* = \mathbf{S}(\mathbf{X}_*,\mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*,\mathbf{X})\Big(\mathbf{S}(\mathbf{X},\mathbf{X}) + \Big(\mathbf{diag}(\sigma_\mathbf{I})\Big)^\mathbf{2}\Big)^{-1}\mathbf{K}(\mathbf{X},\mathbf{X}_*)
\end{aligned}
\tag{3.44}
$$

Using Expectation of Taylor series for covariance matrix

For the second case, like the former case, first, the joint distribution and then the predictive distribution is obtained as given below,

$$
P(\mathbf{y},\mathbf{f}_*|\mathbf{X},\mathbf{X}_*,\Phi) \sim \mathcal{N}\Big(\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{S}(\mathbf{X},\mathbf{X}) + \Big(\mathbf{diag}(\sigma_\mathbf{I})\Big)^\mathbf{2} & \mathbf{S}(\mathbf{X},\mathbf{X}_*) \\ \mathbf{S}(\mathbf{X}_*,\mathbf{X}) & \mathbf{S}(\mathbf{X}_*,\mathbf{X}_*)\end{bmatrix}\Big) \tag{3.45}
$$

$$
\begin{aligned}
&P(\mathbf{f}_*|\mathbf{y},\mathbf{X},\mathbf{X}_*,\Phi) \sim \mathcal{N}(\boldsymbol{n_*},\mathbf{V}_*) \\
&\quad where \qquad \boldsymbol{n_*} = \mathbf{S}(\mathbf{X}_*,\mathbf{X})\Big(\mathbf{S}(\mathbf{X},\mathbf{X}) + \Big(\mathbf{diag}(\sigma_\mathbf{I})\Big)^\mathbf{2}\Big)^{-1}\mathbf{y}, \\
&\qquad\qquad\quad \mathbf{V}_* = \mathbf{S}(\mathbf{X}_*,\mathbf{X}_*) - \mathbf{S}(\mathbf{X}_*,\mathbf{X})\Big(\mathbf{S}(\mathbf{X},\mathbf{X}) + \Big(\mathbf{diag}(\sigma_\mathbf{I})\Big)^\mathbf{2}\Big)^{-1}\mathbf{S}(\mathbf{X},\mathbf{X}_*)
\end{aligned}
\tag{3.46}
$$

where $\mathbf{S}(\mathbf{X},\mathbf{X})$, $\mathbf{S}(\mathbf{X}_*,\mathbf{X}) = \mathbf{S}(\mathbf{X},\mathbf{X}_*)^T$, and $\mathbf{S}(\mathbf{X}_*,\mathbf{X}_*)$ are the approximated covariances between training data and themselves, the testing and training dataset, and the test dataset and themselves respectively, whose entries are derived using Eq.(3.23).

3.5 Examples and Results

3.5.1 Numerical example

For a first example, we consider the following nonlinear function in Eq. (3.47). The signal input was generated as a linearly spaced vector, then we corrupted the input signal by a normal distribution with standard deviation 0.2 and mean zero. The corresponding target value function is calculated by the equation below,

$$
f(x) = sin(x) + 0.5 * sin(2.7 * x) \tag{3.47}
$$

In order to check the performance of the proposed model, we also corrupt the target function values of the training data set by a Gaussian mixture measurement noise where 70% of the noise data were generated from $\mathcal{N}(0, 0.173^2)$, and the remaining 30% of the noise data were generated from$N(0, 1.225^2)$. Training dataset includes 100 samples, and testing dataset includes 1000 samples. The generated training dataset was modelled with the proposed robust GP with noisy input (NIGPGM) and predicted value was compared with validation data, as well as the existing methods in the literature. The noise hyper-parameters including input noise parameters, output noise parameters are computed by the proposed method and other methods and the results are presented in Table 3.1,

Table 3.1: Comparing the estimated hyper-parameters using three methods with their true value

Hyper-parameters / Methods	α_1	α_1	σ_1	σ_2	σ_x
True value	0.3	0.7	1.225	0.173	0.2
Standard GP	1	0	0.7425	Not available	Not available
NIGP	1	0	0.6816	Not available	0.2553
NIGPGMM	0.2307	0.7693	1.421	0.238	0.1758

We can observe that the estimated parameters from the proposed method, are close to the true value of injected input and output noise hyper-parameters. We compared the performance of the proposed method and that of two methods including standard GP and standard GP with noisy input. Table 3.2 shows the performance of these methods in terms of mean absolute error (MAE) and root mean square error (MRSE).

Table 3.2: Prediction Performance using three methods

Method	MAE	MRSE
GP	0.1673	0.1981
NIGP	0.1392	0.1619
NIGPGM	**0.0772**	**0.0900**

Fig.(3.3) and Fig.(3.4) present the mean prediction of the proposed method (NIGPGM), GP, and NIGP. It is clear that the proposed method outperforms the other two methods.

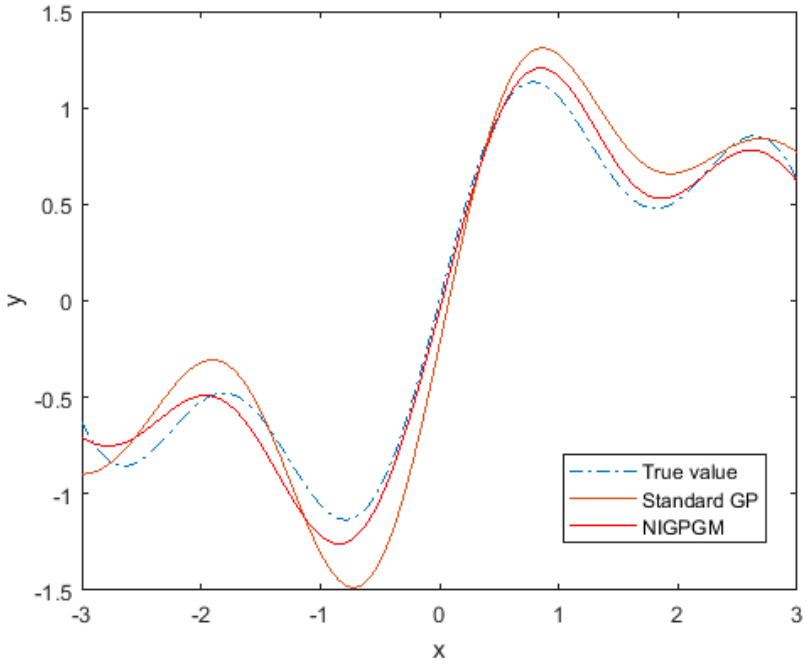

Figure 3.3: The mean prediction of NIGPGM and GP methods on the validation dataset

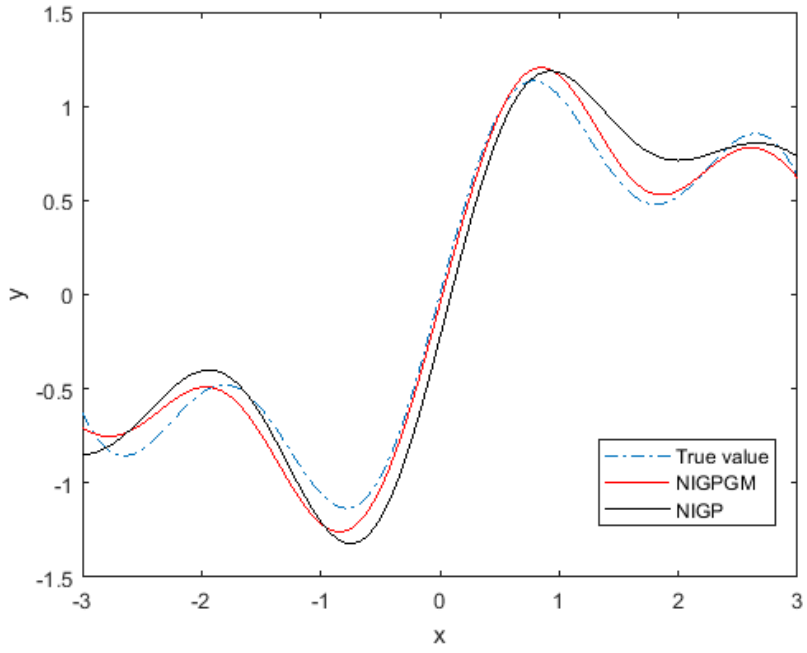

Figure 3.4: The mean prediction of NIGPGM and NIGP methods on the validation dataset

3.5.2 Simulation

As a second example, we consider the CSTR simulation example from Chapter 2. In this simulation example, we attempt to find the underlying function which maps the input,coolant flow rate (F_C), to one of the outputs, i.e. product concentration of component A (C_A). Here, we injected a Gaussian noise distribution $\mathcal{N}(0, 0.2^2)$ to the input signal. Then, this noisy input is used as an input to the CSTR simulation to get the simulated value of the output. The output signal generated by the simulation was corrupted with the following Gaussian mixture noise:

$$\epsilon \sim \begin{cases} \mathcal{N}(0, 4.0000e-5) & w.p \quad 0.7 \\ \mathcal{N}(0, 0.009) & w.p \quad 0.3 \end{cases} \tag{3.48}$$

The input and output pairs for the training part are illustrated in Fig.(3.5).

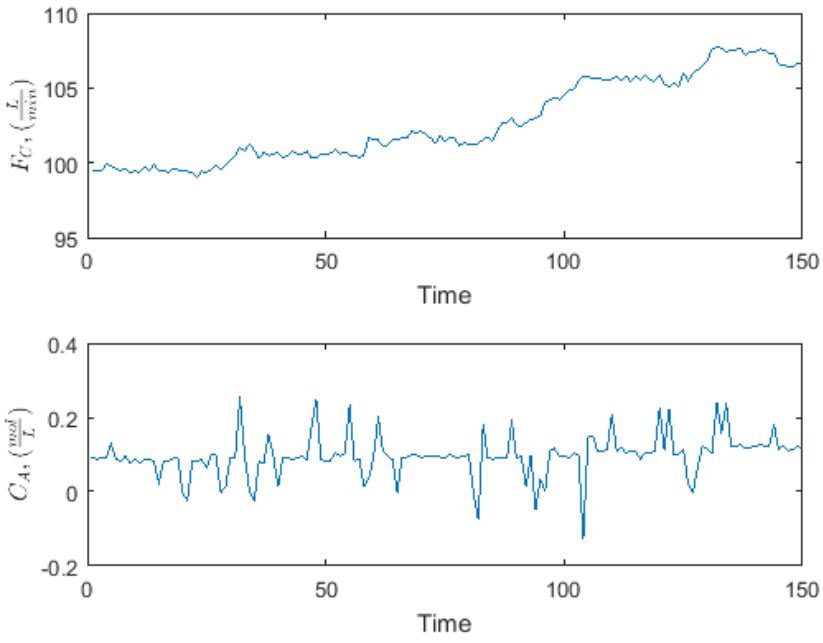

Figure 3.5: The noisy input and output dataset used for training the CSTR model

The prediction performance of the proposed model (NIGPGM) is presented in Fig.(3.6). Table 3.3 presents the magnitudes of MAE for the standard GPR and the proposed NIGPGM model, which clearly verify that the proposed method has a better prediction performance. Further, the estimated hyper-parameters are very

close to their true value as well which are as follows,

$$[\alpha_1, \alpha_2, \sigma_1^2, \sigma_2^2, \sigma_x] = [0.7257, 0.2743, 0.01866, 2.3141, 0.2777]$$

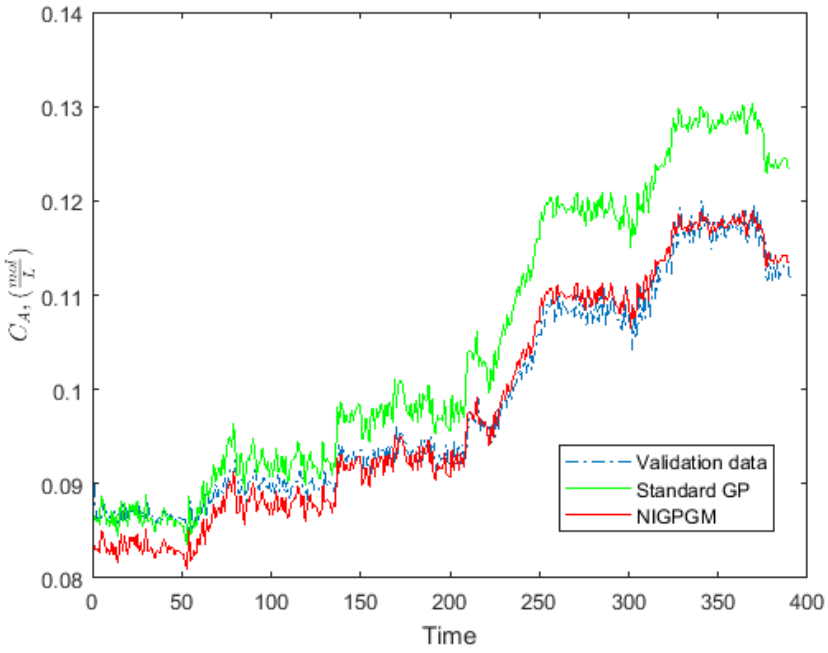

Figure 3.6: The mean prediction of NIGPGM and GP methods on CSTR dataset

Table 3.3: Prediction Performance of CSTR

Method	MAE	MSE
GP	0.00628	0.00628
NIGPGM	**0.0018817**	**0.0022813**

3.6 Conclusion

In this Chapter, we proposed a new robust GPR model in which the input noise is modelled with a Gaussian distribution and the outputs are corrupted by a mixture of two Gaussian distributions noise to model processes affected by both the input and output noise. Further, using two different approaches we approximated the GP prior with a new GP model whose kernel function is dependent on both the input noise and the output noise hyper-parameters. Then we presented a method based on EM algorithm to fit the parameters of the proposed model, thereby the predictive

distribution model is constructed. Numerical and simulation examples have been used to demonstrate the effectiveness of the proposed method.

Chapter 4

Modelling of Dynamical Systems with Robust Gaussian Process Regression

In time series analysis, since data is collected over time and is of sequential nature, the main goal is to extrapolate previously observed values into the future values and to characterize the dynamical properties of the data. In this chapter, a robust Gaussian process is utilized as a novel approach to identify nonlinear dynamic systems. Using a robust Gaussian process regression (GPR) model with a mixture of two Gaussians noise model, presented in the two previous chapters, the iterative one step ahead prediction is made. Within this framework, it is assumed that the underlying function that maps regressors to the current output is modelled by a robust identification method to learn the true dynamics of the system that has been corrupted by outliers. In order to demonstrate the effectiveness of the proposed method, a simulated data set and a Mackey-Glass chaotic time series data set are investigated.

4.1 Introduction

Dynamic system identification describing mathematical models that govern the systems' dynamics given the observed data has attracted the attention of control system engineers for many years. Depending on the form of the systems' dynamics, linear or nonlinear system identification methods are employed to describe the behaviour of the systems [Billings, 2013]. Linear system identification is utilized for systems that satisfy the superposition property. It is commonly represented by the autoregres-

sive moving average with exogenous input (ARMAX) model, and other models that are the subset of the ARMAX, such as AR (autoregressive), ARMA (autoregressive moving average), and ARX(autoregressive with exogenous input) models. ARMA model, which was first introduced in Whitle [1951], is composed of both the autoregressive(AR) and moving average (MA) models. Interested readers are referred to Box et al. [1970], Ljung and Söderström [1983], Ljung [1987], and Soderstrom and Stoica [1989] for the detailed description of these identification methods.

Nonlinear system identification has been widely studied with a focus on nonlinear systems that do not satisfy the superposition principle. An extensive class of nonlinear systems can be represented by the nonlinear autoregressive moving average with exogenous input (NARMAX) model, which was first introduced in Billings and Leontaritis [1980]. Other representations for nonlinear systems, such as Volterra series, block structured, and neural network models can be viewed as subsets of NARMAX. Local modelling approach has also been utilized in the nonlinear dynamic modeling, wherein a complex nonlinear system is divided into subsystems that are independently modelled [Johansen et al., 1999]. Most traditional approaches to nonlinear system identification are based on parametric methods. Furthermore, non-parametric probabilistic approaches, namely, fuzzy model [Takagi and Sugeno, 1985] and neural network [Narendra and Parthasarathy, 1990] have been extensively used as the nonlinear modelling approaches. Back propagation neural network, which has been used by ([Lapedes and Farber [1987];Principe et al. [1992]) in nonlinear system identification of time series, has produced more accurate results compared to traditional methods.

The Gaussian process (GP) model, one of the non-parametric alternatives to static modelling, outperforms the other non-parametric models according to a study by [Rasmussen, 1999]. In the context of nonlinear dynamic modelling, GP has also been explored in several studies. For instance, due to several limitations of multiple local modelling techniques in some applications, Murray-Smith et al. [1999] modelled a nonlinear problem by proposing a multiple local models approach including non-parametric GPs as a prior in off-equilibrium regimes and linear sub-models in areas around equilibrium. Leith et al. [2002] has also used a hybrid approach, which is a combination of a local linear modelling approach and a non-parametric GP, to reduce

the off-equilibrium problem. They have mentioned that a non-minimal realization might be required to accurately capture the dynamics of a system. As an extension to Leith et al. [2002], Ažman and Kocijan [2011] explored the feasibility of the method for higher order dynamic systems. Further, Gregorcic and Lightbody [2007] proposed a local model network based-algorithm for nonlinear system identification, wherein a global GP with a nonlinear kernel function is used to identify the structure.

As shown in O'HAGAN [1992], the derivative of a GP is also a GP. Solak et al. [2003] used this property to model nonlinear dynamic systems by incorporating the derivative and function value observations. Girard et al. [2003b] and Damianou and Lawrence [2015] developed different approximation methods for learning a standard GP with uncertain inputs in which the uncertainty across multi-step ahead prediction was propagated. Inspired by GP latent variable model (GPLV), Wang et al. [2006] proposed a GP dynamical model to capture the dynamics of high dimensional data. All of the previously mentioned works utilized a structure for creating an internal memory to exhibit temporal patterns, i.e. recurrent models. Similar to recurrent neural networks (RNNs), Mattos et al. [2015a] presented a recurrent GP (RGP) and developed recurrent variational Bayes (REVARB) framework that propagates the uncertainty across the RGP states. Following Mattos et al. [2015a], a recurrent deep GP-based approach was proposed for identification of dynamic system in Mattos et al. [2017]. Their proposed approach introduced two groups of networks containing multilayer and recurrent methods in a novel interconnected algorithm to identify the dynamics of a system. A similar work has been previously published by Narendra and Parthasarathy [1990] in the neural network context.

State-space models (SSMs) and nonlinear auto-regressive with exogenous input (NARX) models are widely used in the context of nonlinear system identification. SSM time-series has been studied by Frigola et al. [2014] wherein variational sparse GPs were employed to approximate a tractable posterior over nonlinear dynamical systems. Further, Kocijan et al. [2005] modelled the dynamics of the system by employing the standard GP model with Gaussian likelihood in NARX structure model as another well-known time series.

Most aforementioned works assumed that the measurement noise adopts a Gaussian distribution. However, in reality, observations which have been corrupted by

outliers or other disturbances contain non-Gaussian noise. In order to accommodate outliers, a heavy-tailed distribution such as the Gaussian mixture distribution, t-distribution, and Laplace distribution is employed as a noise likelihood to render the model robust to outliers. Unlike standard GP, the inference of the robust GPR models is intractable and several approximate inference methods have been developed in the literature, including Kuss [2006] and Tipping and Lawrence [2005]. Moreover, Mattos et al. [2015b] employed these robust GPR models with t-distribution and Laplace distribution likelihood in nonlinear dynamical system identification and compared the results with the standard GP-based approach. The NARX structure was changed in Mattos et al. [2016] and the latent autoregressive Gaussian process model (GPLARX) that considers the additional uncertainty caused by feed-backing the noisy input into the model was proposed.

In this chapter, we investigate the identification of nonlinear dynamical systems with the NARX structure that uses a robust GPR model with a Gaussian mixture noise model in order to obtain a more accurate solution for the identification of the dynamical systems with outliers. By assuming a mixture of two Gaussian noises, the regular noise is captured by the Gaussian distribution with smaller variance, and outliers or other disturbances are modelled by the second normal distribution with higher variance. The inference of this robust GPR is analytically intractable, and the approximate learning for this regression model was established in Chapter 2 of this thesis. Further, we employ the learning approach from Chapter 3 in the noisy input case to consider the uncertainty resulting from feed-backing the noisy data into the model. Finally, we compare the results with standard GP-based identification in order to illustrate the improvements.

The rest of this chapter is organized as follows; the problem statement is described in Section 4.2. In Section 4.3, we formulate the proposed robust GPR and deploy it to the NARX model. In Section 4.4, we report and discuss the simulation results for a simulated data set, as well as Mackey-Glass chaotic time series data obtained using the proposed robust system identification with the existing system identification method based on the standard GP. Conclusions are provided in Section 4.5.

4.2 Basic problem description

Consider the learning task of an underlying temporal pattern which has been structured by a NARX model comprising a mapping from the regressor to the observed output. This dynamic system model relates the current output to previously observed outputs and control inputs by an unknown nonlinear function. The proposed model structure for the set of data can be written as follows:

$$\begin{aligned} & y(t) = q(\boldsymbol{\vartheta}(t)) + \varepsilon(t) \\ \textit{where} \quad & \boldsymbol{\vartheta}(t) = [y(t-1), u(t-1), y(t-2), u(t-2), \ldots, y(t-L_y), u(t-L_u)], \end{aligned} \tag{4.1}$$

L_y and L_u are the number of delayed outputs and control inputs, respectively; $\boldsymbol{\vartheta}(t)$ denotes the regressor which is in principle comprised of L_y delayed outputs and L_u delayed control inputs prior to time t; $y(t)$ denotes the current output; and $q(.)$ represents the underlying nonlinear function. The measurement noise, $\varepsilon(t)$, of this dynamic model follows a mixture of two Gaussian distributions as,

$$\varepsilon(t) \sim \begin{cases} \mathcal{N}\left(0, \sigma_1^2\right) & w.p.\ \pi_1 \\ \mathcal{N}\left(0, \sigma_2^2\right) & w.p.\ \pi_2 \end{cases} \tag{4.2}$$

where σ_1^2 denotes the variance of the regular noise distribution with a small value and σ_2^2 with a relatively large value presents the variance of the outliers distribution; π_j denotes the probability of the occurrence of outliers or regular noises, resulting in $\sum_{j=1}^{2} \pi_j = 1$. By using the aforementioned model for the noise likelihood, the potential presence of outliers is addressed, which makes the identification task robust to large random errors. In this setting, we wish to find the underlying process $\mathbf{q}$ given a time-series up to time n:

$$\begin{aligned} & \mathcal{D} = \{\mathbf{y}, \mathbf{V}\} \\ & \mathbf{V} = \{\boldsymbol{\vartheta}(t) | t = 1, 2, \ldots, n\} \\ & \mathbf{y} = \{y(t) | t = 1, 2, \ldots, n\}. \end{aligned} \tag{4.3}$$

where the regressor vector is assumed to be d-dimensional: $\boldsymbol{\vartheta}(t) \in \mathcal{R}^d$. We place a GP as a prior knowledge on $\mathbf{q}(\mathbf{V}) = \{q(\boldsymbol{\vartheta}(1)), q(\boldsymbol{\vartheta}(2)), \ldots, q(\boldsymbol{\vartheta}(n))\}$ function. Fig.(4.1) illustrates the graphical model for the proposed robust system identification using GPR with a mixture of two Gaussians noise model.

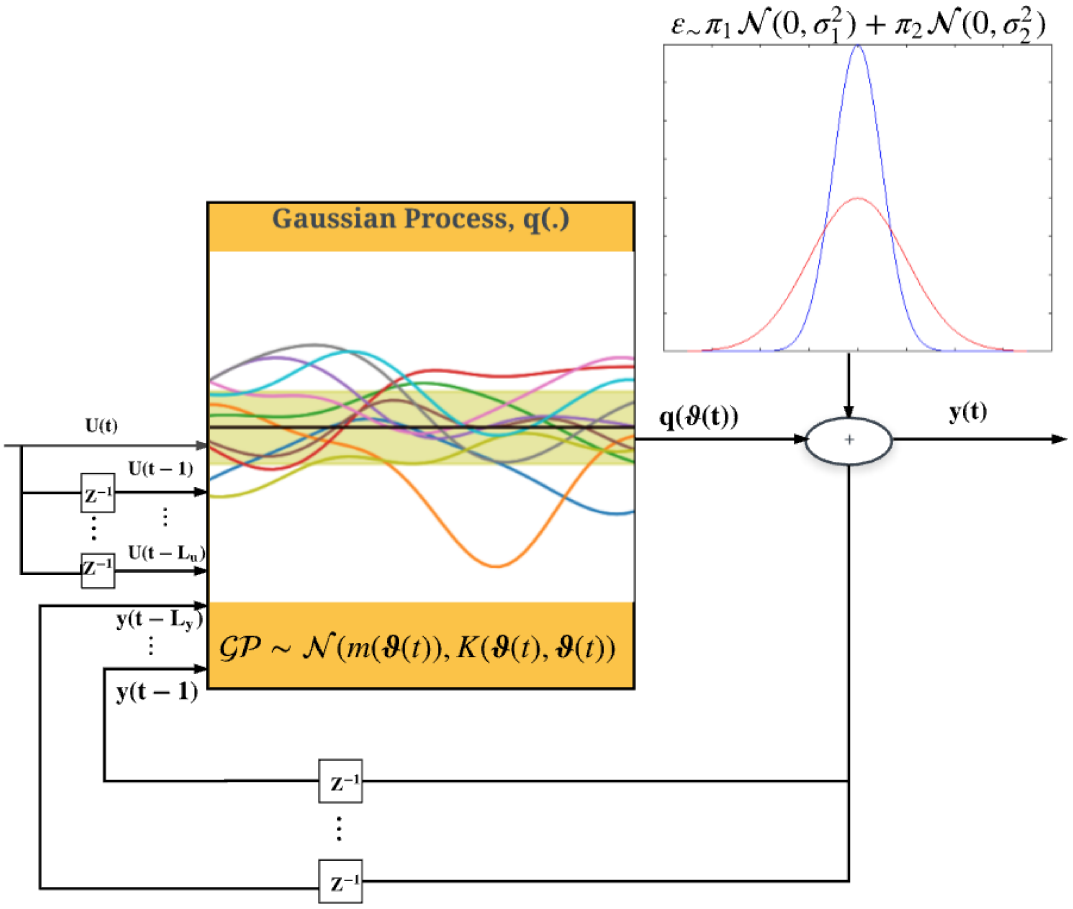

Figure 4.1: A schematic of the proposed robust system identification using GPR with a mixture of two Gaussian noises

In the proposed method, the conditional distribution of $p(\mathbf{q}(\mathbf{V})|\mathbf{V})$ is assumed to follow a multivariate Gaussian which is completely characterized by its mean function $\mathbf{m}(\mathbf{V})$ and the covariance function $\mathbf{K}(\mathbf{V}, \mathbf{V})$. Similar to the previous chapters, the square exponential (SE) kernel is formulated as,

$$k_{SE}(\boldsymbol{\vartheta}(i), \boldsymbol{\vartheta}(j)) = \sigma_f^2 \ \exp\Big(-\frac{1}{2}(\boldsymbol{\vartheta}(i) - \boldsymbol{\vartheta}(j))^T \boldsymbol{\Lambda}^{-1} (\boldsymbol{\vartheta}(i) - \boldsymbol{\vartheta}(j)) \Big) \tag{4.4}$$

where signal variance is denoted by σ_f^2, and the length-scale diagonal matrix is presented by $\boldsymbol{\Lambda} = diag([{l_1}^2, {l_2}^2, .., {l_d}^2])$.

In finding the underlying function, $q(.)$, a simplistic approach is to view this identification task as an extension to the nonlinear regression problem which has been

discussed in Chapters 2 and 3. By modelling this dynamic system using the proposed robust GP with a Gaussian mixture noise model, we can make multi-step ahead predictions for some unobserved data based on one step ahead prediction iteratively. These steps are presented in detail in the next section.

4.3 K-steps ahead prediction based on the dynamic system identification using the proposed robust GPR

In this section, the K-step ahead prediction through the iteration of each successive prediction for two different cases are discussed: (i) We only feed back the mean prediction,(ii) We feed back the predictive probability density to involve the variance prediction as well as the mean prediction. As mentioned in Section 4.2, $y(t) = q(\boldsymbol{\vartheta}(t)) + \epsilon(t)$ where $\boldsymbol{\vartheta}(t) = [y(t-1), \ldots, y(t-L_y), u(t-1), \ldots, u(t-L_u)]$. For simplicity in notation, $q(\boldsymbol{\vartheta}(t))$ will be denoted by $q(t)$. As the time series is assumed to be known up to time $t = n$ and $L_y > L_u$, the identification data comprised of the input (V) and the corresponding output (y) for the NARX model, based on GPR with a Gaussian mixture noise model, (GPGM-NARX) can be formulated as,

$$\mathbf{y} = \begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(L_y+1) \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} y(n-1) & \ldots & y(n-L_y) & U(n-1) & \ldots & U(n-L_u) \\ y(n-2) & \ldots & y(n-L_y-1) & U(n-2) & \ldots & U(n-L_u-1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y(L_y) & \ldots & y(1) & U(L_y) & \ldots & U(L_y-L_u+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\vartheta}(n) \\ \boldsymbol{\vartheta}(n-1) \\ \vdots \\ \boldsymbol{\vartheta}(L_y+1) \end{bmatrix}$$

In the remainder of this chapter, we assume a GP as a prior on $\mathbf{q} = [q(1), \ldots, q(n)]$ with zero mean (as the prior mean function) and the squared exponential kernel function (as the prior covariance function). Since the noise model follows a mixture of two Gaussian distributions to account for the outliers in industrial data, we use the formulation from Chapter 2. Using the above datasets, we train the hyper-parameters of Gaussian process and mixture noise parts. After learning the complete

set of hyper-parameters, the predictive distribution of the unobserved output $y(n+1)$ corresponding to the input $\boldsymbol{\vartheta}(n+1)$ can be derived as,

$$P(y(n+1)|\mathbf{V},\mathbf{y},\boldsymbol{\vartheta}(n+1)) \sim \mathcal{N}\Big(m(\boldsymbol{\vartheta}(n+1)), S(\boldsymbol{\vartheta}(n+1))\Big) \tag{4.5}$$

4.3.1 Naive approach

In the first case, we only consider that the mean predictions, $m(\boldsymbol{\vartheta}(n+1))$, are fed back into the model. Using the estimated value, $\hat{y}(n+1) = m(\boldsymbol{\vartheta}(n+1))$, we construct a new regressor vector as follows,

$$\begin{aligned}\boldsymbol{\vartheta}(n+2) &= [\hat{y}(n+1), y(n), \ldots, y(n-L_y+2), U(n+1), U(n), \ldots, U(n-L_u+2)] \\ &= [m(\boldsymbol{\vartheta}(n+1)), y(n), \ldots, y(n-L_y+2), U(n+1), U(n), \ldots, U(n-L_u+2)]\end{aligned} \tag{4.6}$$

to obtain the predictive distribution of $P(y(n+2)|\mathbf{V},\mathbf{y},\boldsymbol{\vartheta}(n+2)) \sim \mathcal{N}\Big(m(\boldsymbol{\vartheta}(n+2)), S(\boldsymbol{\vartheta}(n+2))\Big)$. Again, the mean of the predictive distribution is utilized as the point estimation of $\hat{y}(n+2)$ which is derived by the formulation from Chapter 2. This step is repeated in order to obtain the point estimation for $y(n+K)$ which is the K-step ahead prediction ($K > L_y$). The regressor for this step, similar to the previous steps, will be derived using the formulation for mean prediction from Chapter 2 as,

$$\begin{aligned}\boldsymbol{\vartheta}(n+K) &= [\hat{y}(n+K-1), \hat{y}(n+K-2), \ldots, \hat{y}(n+K-L_y), U(n+K-1), \ldots)] \\ &= [m(\boldsymbol{\vartheta}(n+K-1)), m(\boldsymbol{\vartheta}(n+K-2)), \ldots, m(\boldsymbol{\vartheta}(n+K-L_y)), U(n+K-1)\ldots]\end{aligned} \tag{4.7}$$

This regressor is used to get the predicted value for $y(n+K)$ as,

$$\begin{aligned}&P(y(n+K)|\boldsymbol{\vartheta}(n+K),\mathbf{V},\mathbf{y},\theta) \sim \mathcal{N}(m(\boldsymbol{\vartheta}(n+K)), S(\boldsymbol{\vartheta}(n+K))) \\ &\textit{where} \quad m(\boldsymbol{\vartheta}(n+K)) = \mathbf{K}(\boldsymbol{\vartheta}(\mathbf{n}+\mathbf{K}),\mathbf{V}) \underbrace{\Big(\mathbf{K}(\mathbf{V},\mathbf{V}) + \Big(\mathbf{diag}(\sigma_{\mathbf{I}})\Big)^{\mathbf{2}}\Big)^{-1}\mathbf{y}}_{\alpha}, \\ &\qquad S(\boldsymbol{\vartheta}(n+K)) = \mathbf{K}(\boldsymbol{\vartheta}(\mathbf{n}+\mathbf{K}),\boldsymbol{\vartheta}(\mathbf{n}+\mathbf{K})) \\ &\qquad\qquad - \mathbf{K}(\boldsymbol{\vartheta}(\mathbf{n}+\mathbf{K}),\mathbf{V})\Big(\mathbf{K}(\mathbf{V},\mathbf{V}) + \Big(\mathbf{diag}(\sigma_{\mathbf{I}})\Big)^{\mathbf{2}}\Big)^{-1}\mathbf{K}(\mathbf{V},\boldsymbol{\vartheta}(\mathbf{n}+\mathbf{K}))\end{aligned} \tag{4.8}$$

4.3.2 Exact Approach

In this case, we feed back the predictive distribution into the model to propagate uncertainty resulting from each successive prediction. Then, the predictive distribution of $y(n+1)$ similar to the former case, can be obtained based on the formulation provided in Chapter 2,

$$P(y(n+1)|\mathbf{V},\mathbf{y},\boldsymbol{\vartheta}(n+1)) \sim \mathcal{N}\Big(m(\boldsymbol{\vartheta}(n+1)), S(\boldsymbol{\vartheta}(n+1))\Big) \tag{4.9}$$

By using this predictive distribution, we construct a new regressor vector for the next prediction step as,

$$\begin{aligned}\boldsymbol{\vartheta}(n+2) &= [\hat{y}(n+1), y(n), \ldots, y(n-L_y+2), U(n+1), U(n), \ldots, U(n-L_u+2)] \\ &[\mathcal{N}\Big(m(\boldsymbol{\vartheta}(n+1)), S(\boldsymbol{\vartheta}(n+1))\Big), y(n), \ldots, y(n-L_y+2), U(n+1)\ldots, U(n-L_u+2)]\end{aligned} \tag{4.10}$$

where $\boldsymbol{\vartheta}(n+2)$ is the noisy input which is fed back into the model. Since we wish to predict its corresponding output, we should be able to derive the predictive distribution of $p(y(n+2)|\mathbf{V},\mathbf{y},\boldsymbol{\vartheta}(n+2))$ for the normally distributed input, $\boldsymbol{\vartheta}(n+2)$, which can be represented by,

$$\boldsymbol{\vartheta}(n+2) = \mathbf{x}(n+2) + \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}}) \tag{4.11}$$

where $\mathbf{x}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ denote noise free input and the covariance matrix of the input noise, respectively, and in the second step both can be obtained as,

$$\mathbf{x}(n+2) = \begin{bmatrix} m(\boldsymbol{\vartheta}(n+1)) & y(n) & \ldots & y(n-L_y+2) & U(n+1) & \ldots & U(n-L_u+2)\end{bmatrix}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \begin{bmatrix} S(\boldsymbol{\vartheta}(n+1)) & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}$$

The approximated predictive distribution for $y(n+2)$ is derived by the results from Chapter 3 as follows:

$$p(y(n+2)|\mathbf{V},\mathbf{y},\boldsymbol{\vartheta}(n+2)) \sim \mathcal{N}\Big(m_+(\boldsymbol{\vartheta}(n+2)), S_+(\boldsymbol{\vartheta}(n+2))\Big) \tag{4.12}$$

We iterate these steps in order to obtain the predictive distribution of $y(n+K)$ for K steps ahead in time from n. Its regressor will be formulated as,

$$\begin{aligned}
&\boldsymbol{\vartheta}(n+K) = [\hat{y}(n+K-1), \hat{y}(n+K-2), \ldots, \hat{y}(n+K-L_y), U(n+K-1), \ldots] \\
&[\mathcal{N}\Big(m_+(\boldsymbol{\vartheta}(n+K-1)), S_+(\boldsymbol{\vartheta}(n+K-1))\Big), \ldots \\
&,\mathcal{N}\Big(m_+(\boldsymbol{\vartheta}(n+K-L_y)), S{+}(\boldsymbol{\vartheta}(n+K-L_y))\Big), U(n+K-1), \ldots, U(n+K-L_u)]
\end{aligned} \tag{4.13}$$

In this step, the noise free input and the covariance matrix of the input noise are as follows,

$$\begin{aligned}
\mathbf{x}(n+K) = \Big[&m_+(\boldsymbol{\vartheta}(n+K-1)) \quad m_+(\boldsymbol{\vartheta}(n+K-2)) \quad \ldots \quad m_+(\boldsymbol{\vartheta}(n+K-L_y)) \\
&U(n+K-1) \quad U(n+K-2) \quad \ldots \quad U(n+K-L_u)\Big]
\end{aligned} \tag{4.14}$$

$$\boldsymbol{\Sigma_x} = \begin{bmatrix}
Cov(y(n+K-1), y(n+K-1)) & \ldots & Cov(y(n+K-1), y(n+K-L_y)) \\
\vdots & \ddots & \vdots \\
Cov(y(n+K-L_y), y(n+K-1)) & \ldots & Cov(y(n+K-L_y), y(n+K-L_y)) \\
Cov(U(n+K-1), y(n+K-1)) & \ldots & Cov(U(n+K-1), y(n+K-L_y)) \\
\vdots & \ddots & \vdots \\
Cov(U(n+K-L_u), y(n+K-1)) & \ldots & Cov(U(n+K-L_u), y(n+K-L_y))
\end{bmatrix}$$
$$\begin{bmatrix}
Cov(y(n+K-1), U(n+K-1)) & \ldots & Cov(y(n+K-1), U(n+K-L_u)) \\
\vdots & \ddots & \vdots \\
Cov(y(n+K-L_y), U(n+K-1)) & \ldots & Cov(y(n+K-L_y), U(n+K-L_u)) \\
Cov(U(n+K-1), U(n+K-1)) & \ldots & Cov(U(n+K-1), U(n+K-L_u)) \\
\vdots & \ddots & \vdots \\
Cov(U(n+K-L_u), U(n+K-1)) & \ldots & Cov(U(n+K-L_u), U(n+K-L_u))
\end{bmatrix} \tag{4.15}$$

As the regressor vector is assumed to be d dimensional, the input covariance matrix, $\boldsymbol{\Sigma_x}$, is $d \times d$. Each entry of the input covariance matrix on the main diagonal can be split into two parts, $Cov(y(n+K-i), y(n+K-i))$ for $i = 1, 2, \ldots, L_y$ and $Cov(U(n+K-i), U(n+K-i))$ for $i = 1, 2, \ldots, L_u$ which are derived as below,

$$Cov(U(n+K-i), U(n+K-i)) = Var(U(n+K-i)) = 0 \tag{4.16}$$

$$Cov(y(n+K-i), y(n+K-i)) = Var(y(n+K-i)) = S_+(\boldsymbol{\vartheta}(n+K-i)) \tag{4.17}$$

Based on the NARX model, exogenous input, U, is usually known and deterministic; there is no uncertainty on the external input, and its variance is zero. The covariance between every feed-back output and itself is calculated by the formulation provided in Chapter 3. Further, the covariance between external(exogenous) inputs and the feed-back outputs is derived as,

$$\begin{aligned} &Cov(U(n+K-i), y(n+K-i)) = Cov(y(n+K-i), U(n+K-i)) \\ &= E[y(n+K-i).U(n+K-i)] - E[y(n+K-i)]E[U(n+K-i)] \\ &= U(n+K-i) * E[y(n+K-i)] - U(n+K-i) * E[y(n+K-i)] = 0 \end{aligned} \tag{4.18}$$

In the final step, we obtain the cross-covariance between the feed-back outputs, $Cov(y(n+K-i), y(n+K-j))$ for $i = 1, 2, ..., L_y - 1$ and $j = i+1, 2, ..., L_y$, by computing the covariance between every output and its corresponding regressor, $Cov(y(n+K-i), \boldsymbol{\vartheta}(n+K-i))$. For $i = 1$, we ignore both the last term of the regressor related to the feed-back output and the whole elements to the external inputs in the regressor vector. Similarly, for $i = 2$, the last two terms of the regressor are disregarded. Similarly, for $i = L_y - 1$, we again disregard the last L_y terms of the regressor. In the following, for simplicity in the notation, we utilize $l = K - i$, and derive the cross-covariance as given below,

$$\begin{aligned} &Cov(y(n+l), \boldsymbol{\vartheta}(n+l)) \\ &= E[y(n+l)\boldsymbol{\vartheta}(n+l)] - E[y(n+l)]E[\boldsymbol{\vartheta}(n+l)] \\ &= E[y(n+l)\boldsymbol{\vartheta}(n+l)] - m_+(\boldsymbol{\vartheta}(n+l))\mathbf{X}(n+l) \end{aligned} \tag{4.19}$$

The first expectation term in Eq.(4.19) will be formulated as follows,

$$E[y(n+l).\boldsymbol{\vartheta}(n+l)]$$

$$= \int\int y(n+l).\boldsymbol{\vartheta}(n+l) p(y(n+l).\boldsymbol{\vartheta}(n+l)) dy.d\boldsymbol{\vartheta}(n+l) \tag{4.20}$$

$$= \int\underline{\int y(n+l).\boldsymbol{\vartheta}(n+l) p(y(n+l)|\boldsymbol{\vartheta}(n+l))}.P(\boldsymbol{\vartheta}(n+l))\underline{dy}.d\boldsymbol{\vartheta}(n+l) \tag{4.21}$$

where the underlined terms are the definition of the expected value of $y(n+l)$ w.r.t $p(y(n+l)|\boldsymbol{\vartheta}(n+l))$ which is denoted by $E_{p(y(n+l)|\boldsymbol{\vartheta}(n+l))}[y(n+l)]$,

$$= \int \boldsymbol{\vartheta}(n+l).E_{p(y(n+l)|\boldsymbol{\vartheta}(n+l))}[y(n+l)].P(\boldsymbol{\vartheta}(n+l)).d\boldsymbol{\vartheta}(n+l) \tag{4.22}$$

$$= \int \boldsymbol{\vartheta}(n+l) m(\boldsymbol{\vartheta}(n+l)).P(\boldsymbol{\vartheta}(n+l)).d\boldsymbol{\vartheta}(n+l) \tag{4.23}$$

By deploying the expression of $m(\boldsymbol{\vartheta}(n+l)) = \sum_i \alpha_i \mathbf{K}(\boldsymbol{\vartheta}(n+l), \boldsymbol{\vartheta}(i))$ according to Eq.(4.8), Eq.(4.23) can be further written as,

$$= \sum_i \alpha_i \int \boldsymbol{\vartheta}(n+l) \mathbf{K}(\boldsymbol{\vartheta}(n+l), \boldsymbol{\vartheta}(i)).P(\boldsymbol{\vartheta}(n+l)).d\boldsymbol{\vartheta}(n+l) \tag{4.24}$$

Based on the Gaussian covariance function, $\mathbf{K}(\boldsymbol{\vartheta}(n+l), \boldsymbol{\vartheta}(i)) = \underbrace{(2\pi)^{D/2}|\Lambda|^{1/2}\sigma_f}_{c} N_{\boldsymbol{\vartheta}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda)$, we have;

$$= c \sum_i \alpha_i \int \boldsymbol{\vartheta}(n+l) N_{\boldsymbol{\vartheta}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda).P(\boldsymbol{\vartheta}(n+l)).d\boldsymbol{\vartheta}(n+l) \tag{4.25}$$

and by replacing $p(\boldsymbol{\vartheta}(n+l))$ with $N_{\boldsymbol{\vartheta}(n+l)}(\mathbf{x}(n+l), \Sigma_x)$, Eq.(4.25) is written as,

$$= c \sum_i \alpha_i \int \boldsymbol{\vartheta}(n+l) N_{\boldsymbol{\vartheta}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda).N_{\boldsymbol{\vartheta}(n+l)}(\mathbf{x}(n+1), \Sigma_x).d\boldsymbol{\vartheta}(n+l) \tag{4.26}$$

Using the product of the Gaussian (refer to Appendix), Eq.(4.26) can be further written as,

$$= c \sum_i \alpha_i \int \boldsymbol{\vartheta}(n+l) N_{\mathbf{x}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda + \Sigma_x).N_{\boldsymbol{\vartheta}(n+l)}(d(i), \underbrace{(\Lambda^{-1} + \Sigma_x^{-1})^{-1}}_{D}).d\boldsymbol{\vartheta}(n+l) \tag{4.27}$$

where $d(i) = D(\Lambda^{-1}\boldsymbol{\vartheta}(i) + \Sigma_x^{-1}\mathbf{x}(n+l))$. Then, Eq.(4.27) can be rewritten as below,

$$= c \sum_i \alpha_i N_{\mathbf{x}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda + \Sigma_x) \underbrace{\int \boldsymbol{\vartheta}(n+l).N_{\boldsymbol{\vartheta}(n+l)}(d(i), D).d\boldsymbol{\vartheta}(n+l)} \tag{4.28}$$

where the term in the under-brace in Eq.(4.28) is the expected value of $\boldsymbol{\vartheta}(n+l)$,

$$= c \sum_i \alpha_i N_{\mathbf{x}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda + \Sigma_x) d(i) \tag{4.29}$$

$$= c \sum_i \alpha_i N_{\mathbf{x}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda + \Sigma_x)(\Lambda^{-1} + \Sigma_x^{-1})^{-1}(\Lambda^{-1}\boldsymbol{\vartheta}(i) + \Sigma_x^{-1}\mathbf{x}(n+l)) \tag{4.30}$$

Finally, by substituting the resulting Eq.(4.30) in Eq. (4.19), the cross-covariance matrix terms are given below,

$$Cov(y(n+l), \boldsymbol{\vartheta}(n+l))$$
$$= c \sum_i \alpha_i N_{\mathbf{x}(n+l)}(\boldsymbol{\vartheta}(i), \Lambda + \Sigma_x)(\Lambda^{-1} + \Sigma_x^{-1})^{-1}(\Lambda^{-1}\boldsymbol{\vartheta}(i) + \Sigma_x^{-1}\mathbf{x}(n+l)) - m_+(\boldsymbol{\vartheta}(n+l))\mathbf{x}(n+l) \tag{4.31}$$

where $m_+(\boldsymbol{\vartheta}(n+l))$ is calculated using the results from the previous chapter. However, Girard et al. [2003a] discuss the predictive distribution for the Gaussian distributed testing data with known mean and variance, which can be exactly computed. It is the learning of noisy inputs which is not analytically tractable and we have to resort to some approximations which have been illustrated in the previous chapter. Thus, instead of using the results from Chapter 3, we can use the predictive distribution for the uncertain testing data with known mean and variance and extend them to the case with the mixture of two Gaussians noise model. Below, we derive the exact moments of the predictive distribution for an uncertain input at time step $n+l$, that is, $\boldsymbol{\vartheta}(n+l)$,

$$p(y(n+l)|\mathbf{V},\mathbf{y},\boldsymbol{\vartheta}(n+l)) \sim \mathcal{N}\Big(m_+(\boldsymbol{\vartheta}(n+l)), S_+(\boldsymbol{\vartheta}(n+l))\Big) \tag{4.32}$$

and its moments are formulated as,

$$\begin{aligned} m_+(\boldsymbol{\vartheta}(n+l)) &= \sum_i \alpha_i E_{\boldsymbol{\vartheta}(n+l)}[\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(i))] \\ S_+(\boldsymbol{\vartheta}(n+l)) &= E_{\boldsymbol{\vartheta}(n+l)}[\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(n+l))] \\ &\quad - \sum_{i,j}(K_{ij}-\alpha_i\alpha_j)E_{\boldsymbol{\vartheta}(n+l)}[\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(i))\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(j))] - m_+(\boldsymbol{\vartheta}(n+l))^2 \end{aligned} \tag{4.33}$$

with $\alpha = \Big(\mathbf{K}(V,V) + (diag(\sigma_I))^2\Big)y$. According to the Gaussian Kernel formulation, we can directly obtain $E_{\boldsymbol{\vartheta}(n+l)}[\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(n+l))] = \sigma_f$. Within the derivation of the first moment of the predictive distribution Eq.(4.34), we reformulate the Gaussian kernel function as a normal density and use the product of the Gaussian. The first moment formulation is obtained as,

$$\begin{aligned} m_+(\boldsymbol{\vartheta}(n+l)) &= \sum_i \alpha_i E_{\boldsymbol{\vartheta}(n+l)}[\underbrace{(2\pi)^{d/2}|\Lambda|^{1/2}\sigma_f}_{c} N(\boldsymbol{\vartheta}(n+l)|\boldsymbol{\vartheta}(i),\Lambda)] \\ &= c\sum_i \alpha_i \int N(\boldsymbol{\vartheta}(n+l)|\mathbf{x}(n+l),\Sigma_x).N(\boldsymbol{\vartheta}(n+l)|\boldsymbol{\vartheta}(i),\Lambda)d\boldsymbol{\vartheta}(n+l) \\ &= c\sum_i \alpha_i N(\mathbf{x}(n+l)|\boldsymbol{\vartheta}(i),\Lambda+\Sigma_x) \end{aligned} \tag{4.34}$$

In deriving the predictive distribution variance Eq. (4.35), we utilize the product

of the Gaussian. The below formulation is obtained for the prediction variance:

$$\begin{aligned} S_+(\boldsymbol{\vartheta}(n+l)) &= E_{\boldsymbol{\vartheta}(n+l)}[\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(n+l))] \\ &\quad - \sum_{i,j}(K_{ij}-\alpha_i\alpha_j)E_{\boldsymbol{\vartheta}(n+l)}[\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(i))\mathbf{K}(\boldsymbol{\vartheta}(n+l),\boldsymbol{\vartheta}(j))] - m_+(\boldsymbol{\vartheta}(n+l))^2 \\ &= \sigma_f^2 - c^2\sum_{i,j}(K_{ij}-\alpha_i\alpha_j)N(\boldsymbol{\vartheta}(i)|\boldsymbol{\vartheta}(j),2\Lambda).N(\mathbf{x}(n+l)|\frac{\boldsymbol{\vartheta}(i)+\boldsymbol{\vartheta}(j)}{2},\Sigma_x+\frac{\Lambda}{2}) \\ &\quad - c^2\sum_{i,j}\alpha_i\alpha_j N(\boldsymbol{\vartheta}(i)|\boldsymbol{\vartheta}(j),2(\Lambda+\Sigma_x)).N(\mathbf{x}(n+l)|\frac{\boldsymbol{\vartheta}(i)+\boldsymbol{\vartheta}(j)}{2},\frac{\Sigma_x+\Lambda}{2}) \end{aligned} \tag{4.35}$$

Thus, according to the above derivations, both moments of the predictive distribution are analytically obtained. $m_+(\boldsymbol{\vartheta}(n+1))$ can be replaced with the exact moments derived by equations Eq.(4.34)-Eq.(4.35). Further, during our derivation, the exogenous inputs were assumed to be deterministic. However, the above derivations can be extended for the stochastic inputs as well, and we treat them the same as the feed-back outputs which are also uncertain.

4.4 Examples

4.4.1 Simulated Example

We consider the example 3 in Narendra and Parthasarathy [1990] as our first example which is described by the following equation:

$$y(t+1) = \frac{y(t)}{1+y^2(t)} + u^3(t) \tag{4.36}$$

where $u(t)$ is the input signal and $y(t)$ is the output signal. This model can be considered as the following the NARX model,

$$y(t+1) = f(y(t),u(t)) \tag{4.37}$$

where function $f(.)$ is modelled by a Gaussian Process with zero mean and squared exponential covariance function. The delayed input and output signals are chosen as regressors. We generated the input $u(t)$ using a pseudo-random signal between a specific range and assumed that the initial state of the system is $y(0)=0$. Using the generated input and Eq.(4.36), the output $y(t+1)$ is also generated. In our case, the identification is done by 100 data samples, where the regressor matrix dimension

is 100×2, and the corresponding output vector is 100×1. The output vector was corrupted by a mixture of two Gaussian noises. For generating noise, we assumed that 70 percent of the noise realizations come from a normal distribution with standard deviation 0.05 and the remaining noise realizations are sampled from another normal distribution with standard deviation 0.7 . The dataset was modelled using the robust GP with mixture Gaussian likelihood. After identification, we can observe the noise hyper-parameters: $\alpha = [0.84, 0.16]$ and $\sigma = [0.067; 0.82]$ are very close to parameters of the injected noise. Then, we made a 200-step ahead prediction using both the naive and the exact approaches, and compared the results with standard GP in terms of mean absolute error (MAE), mean squared error (MSE) and log predictive density (LPD). Table 4.1 summarizes the performance criteria for GPGMM and GP using naive approach. In the naive approach, we can observe that the proposed method outperforms the standard GP.

Table 4.1: Prediction Performance using the Naive approach

Method	MAE	MSE	LPD
GP Naive	0.26854	0.11006	0.29405
GPGMM Naive	**0.11122**	**0.036906**	**4.7674**

Fig.(4.2) shows the mean prediction from 1 to 200-step ahead prediction using the naive approach. This means that the mean prediction is fed back as delayed output in constructing the regressor vector. It is clear from Fig.(4.2) that the proposed method has a better prediction than GP, and the squared error for each of the 200 predicted points is illustrated by Fig.(4.3).

The exact approach is implemented for the proposed method (GPGMM) as well as the standard GP method. In both methods, the predictive probability including mean and uncertainty is fed back as delayed output to construct the next step regressor vector. The proposed method also has better performance in exact approach than the standard GP which has been demonstrated in Fig.(4.4).

From comparing Fig.(4.5) and Fig.(4.6), we can notice that the predictive mean calculated from the exact approach is closer to the system than the naive approach. Moreover, the exact approach provides us with more accurate information about uncertainty which shows the level we can trust the mean prediction.

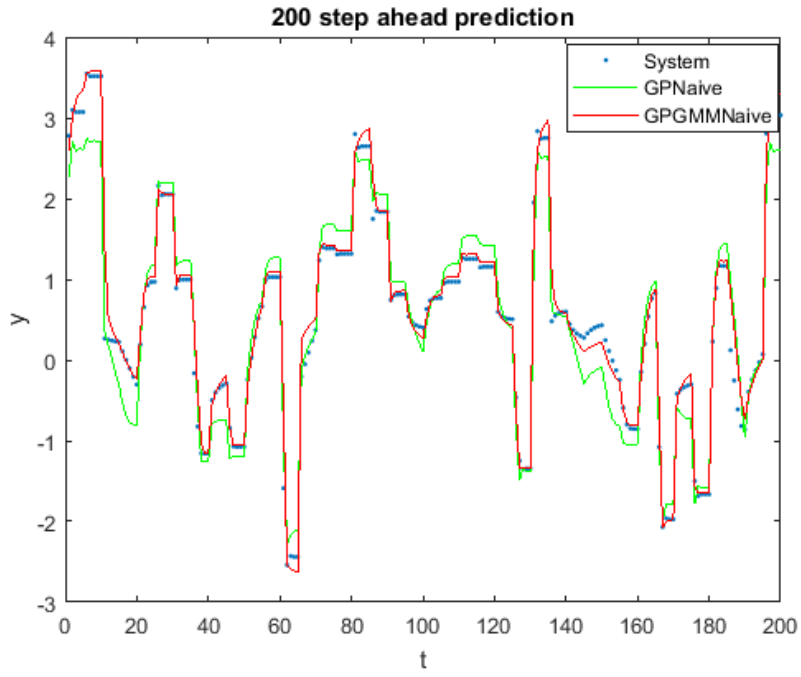

Figure 4.2: The predictive mean using Naive approach

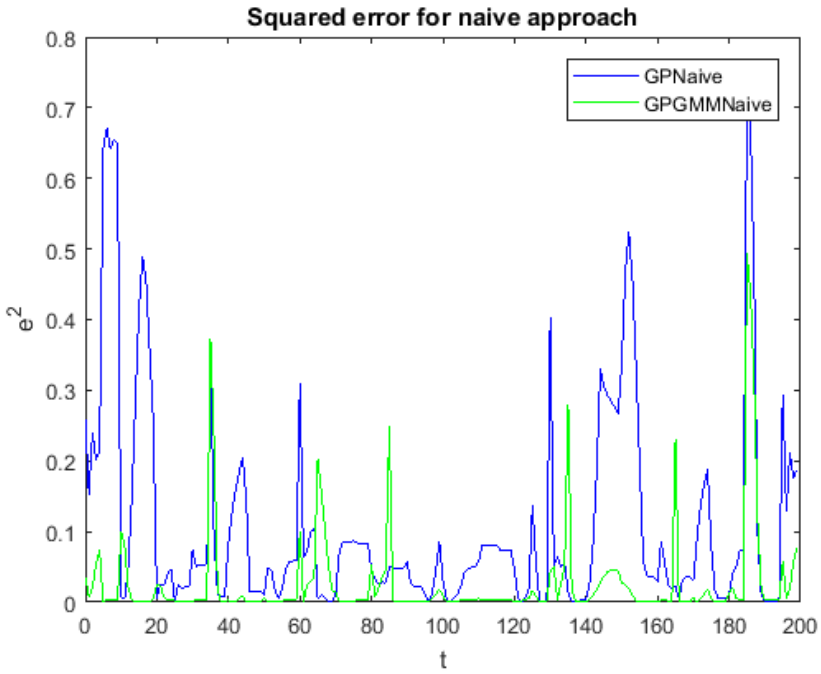

Figure 4.3: The Squared error for each of the 200 predicted points using Naive approach

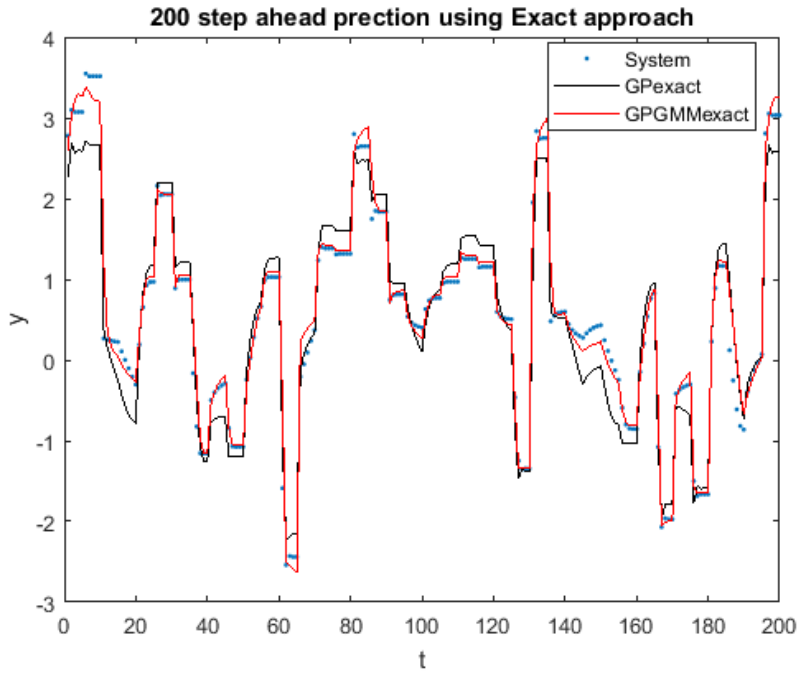

Figure 4.4: The predictive mean using exact approach

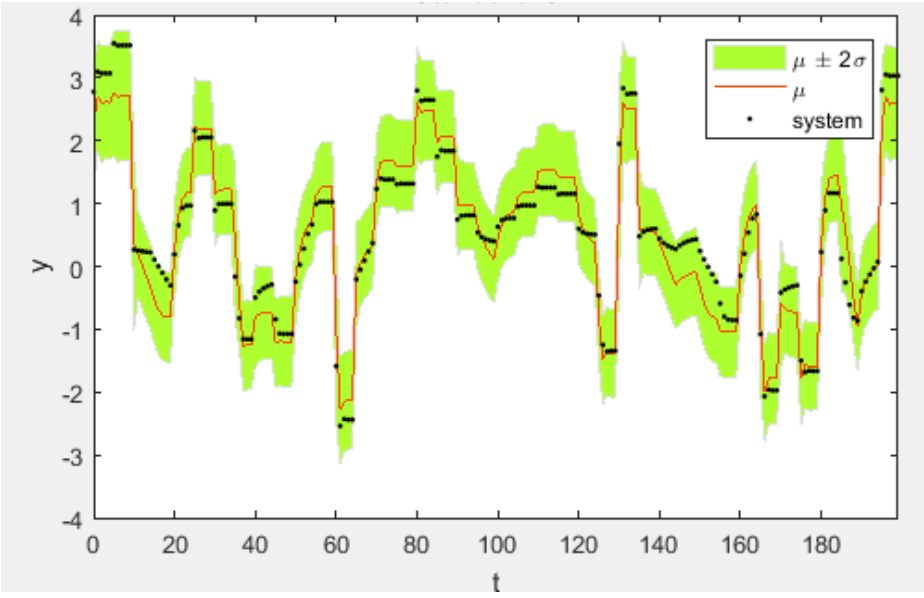

Figure 4.5: The predictive mean and error bars from t+1 to t+200 using GPGMM naive approach

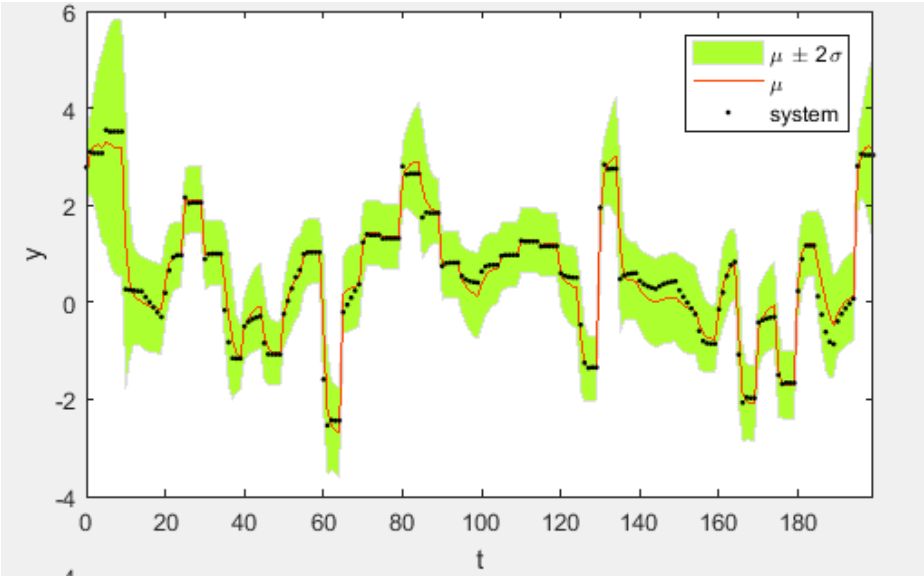

Figure 4.6: The predictive mean and error bars from t+1 to t+200 using GPGMM exact approach

4.4.2 Mackey-Glass chaotic time series

In this last example, we consider Mackey-Glass time series which is a chaotic time-series to investigate the ability of the proposed robust GP for k-step ahead prediction. The Mackey-Glass time-series can be described by the following nonlinear time-delay differential equation,

$$\frac{dy(t)}{dt} = \frac{\beta y(t-\tau)}{1+y(t-\tau)^n} - \lambda y(t) \tag{4.38}$$

where the parameters of this equation: β, τ, n, and λ are real numbers and dependent on their values. This equation shows a different range of chaotic and periodic dynamics. The time delay, τ, should be more than 17 to show chaotic behaviour. Here, we assume $\tau = 17$, $\beta = 0.2$, $\lambda = 0.1$, $n = 10$ and also the initial condition is assumed to be $y(0) = 1.2$. We have used the existing data in MATLAB which has been generated from this time series.

A nonlinear auto-regressive (NAR) model is assumed, suggesting past observations might predict current observations, and has the following form,

$$y(t+1) = f\Big(y(t), y(t-1), ..., y(t-P)\Big) \tag{4.39}$$

where for the above time-series with a time delay equal to 17, the model order is $P = 16$. The output y(t+1) was corrupted by a mixture of two Gaussian noises where 85% of the noise realizations were generated from $\mathcal{N}(0, 0.001)$, and the remaining 15% of the noise realizations were generated from $\mathcal{N}(0, 0.01)$. After constructing the input and the corresponding output pairs which are taken at random from the above time series, we train the model using the robust GP with mixture Gaussian likelihood. After learning the complete set of hyper-parameters, we start to make the 50-step ahead prediction (from time 72 to 121) using both the naive and the exact approaches and compare the result with the standard GP. Table 4.2 indicates the three aforementioned performance criteria for GPGMM and GP using the naive approach whereby we can notice that GPGMM has an improved prediction performance compared with GP.

Table 4.2: Prediction Performance on Mackey Glass time series using the Naive approach

Method	MAE	MSE	LPD
GP Naive	0.05784	0.0042352	-1.2401
GPGMM Naive	**0.023051**	**0.00076725**	**6.7602**

The mean prediction (from 1 to 50-step ahead) using the naive approach by GPGMM and standard GP is presented in Fig.(4.7), and Fig.(4.8) illustrates the squared error for each of 50 steps ahead prediction using both methods. As it is evident from the figures and the table, the proposed method outperforms the GP in terms of prediction performance.

Now, we proceed to make the exact prediction for the proposed method (GPGMM) as well as the standard GP method. Table 4.3, Fig.(4.9) and Fig.(4.10) show the improvement of the proposed method compared to GP using the exact approach.

Table 4.3: Prediction performance on Mackey Glass time series using the exact approach

Method	MAE	MSE	LPD
GP Exact	0.086562	0.0086846	-0.74698
GPGMM Exact	**0.023827**	**0.0009535**	**15.7131**

Fig.(4.11) and Fig.(4.12) present the mean prediction and confidence interval for the proposed method using naive and exact approaches.

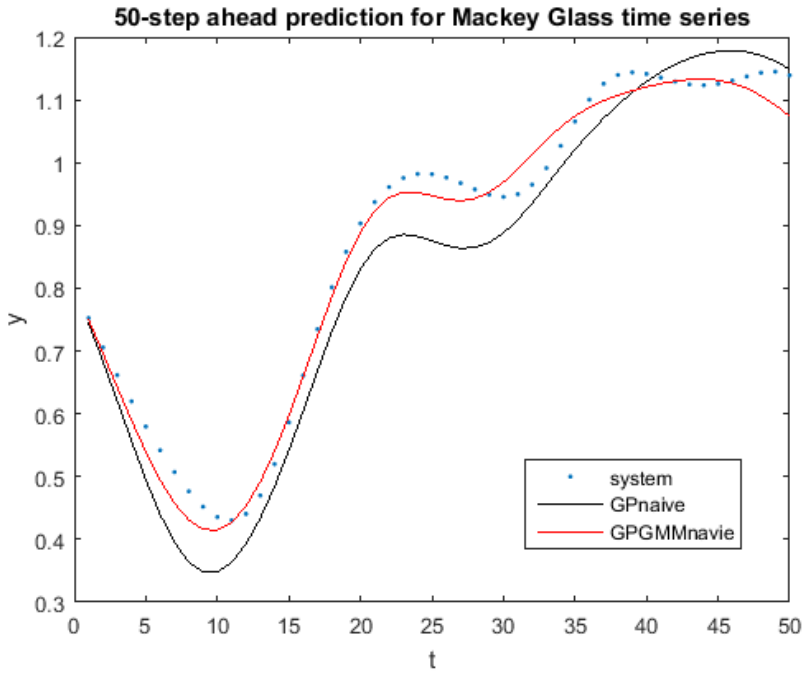

Figure 4.7: The predictive mean using the naive approach

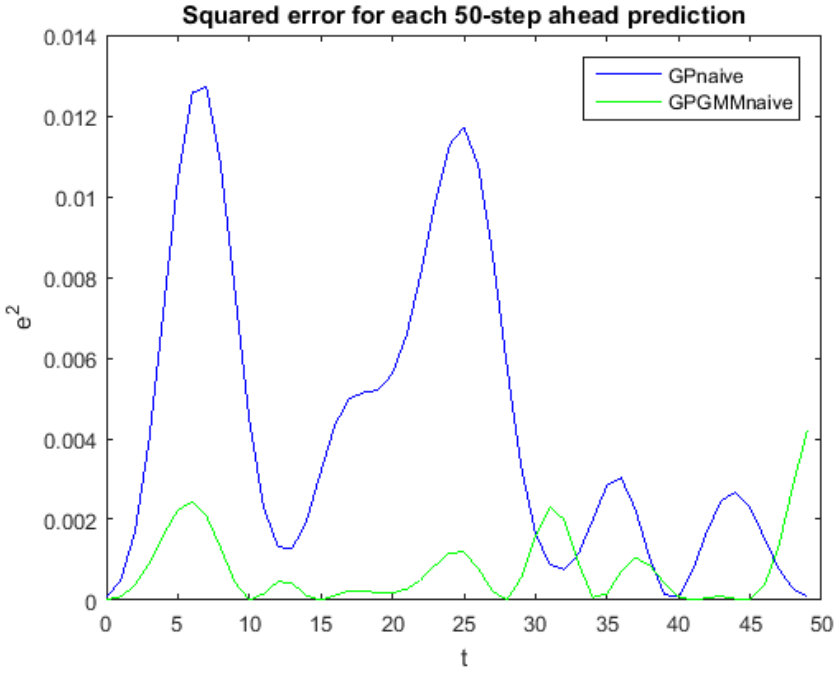

Figure 4.8: The Squared error for each of the 50 steps ahead prediction using the naive approach

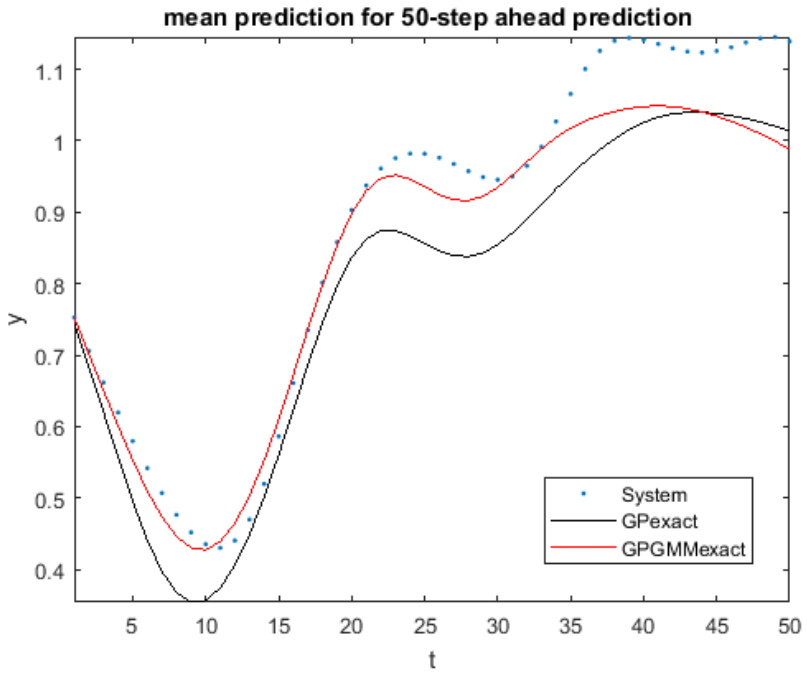

Figure 4.9: The predictive mean using the exact approach

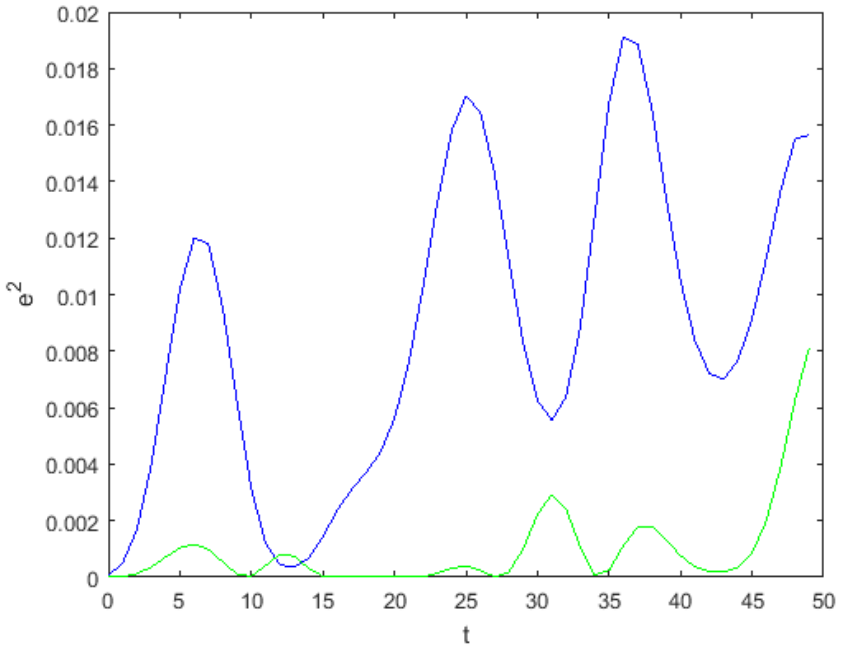

Figure 4.10: The squared error for each of the 50 steps ahead predictions using the exact approach

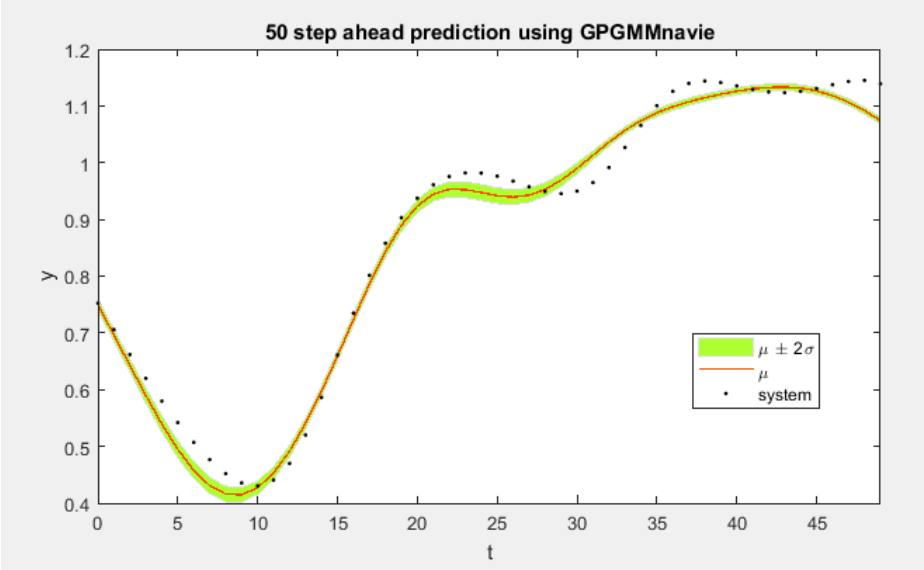

Figure 4.11: The predictive mean and error bars from t+1 to t+50 using GPGMM Naive approach

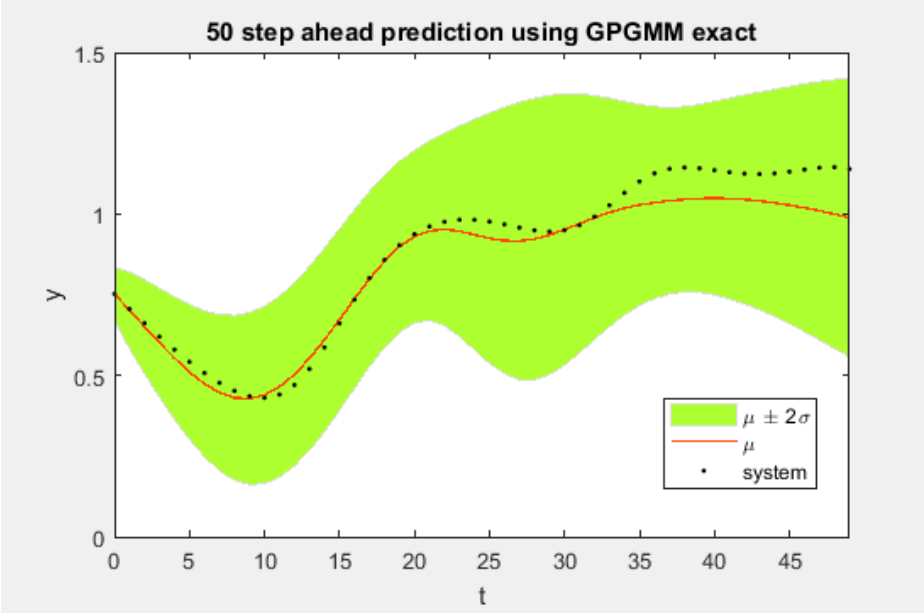

Figure 4.12: The predictive mean and error bars from t+1 to t+50 using GPGMM exact approach

As it is clear from Fig.(4.11), however, the naive approach provides better predictions than the exact approach, and the confidence interval (green area) for the naive approach is too narrow which cannot even include the true value. Hence, the naive approach is over confident about the predictive means in some regions where the performance of predictions is poor. The exact approach represented by Fig.(4.12), unlike the naive approach, provides more informative uncertainty.

4.5 Conclusion

In this chapter, we presented a new robust system identification method based on the proposed GPR with a Gaussian mixture noise model and showed how this robust GP could be utilized for the nonlinear dynamical modelling of the identification data. Specifically, we derived the approximate predictive distribution for an unobserved output at time t+k by the iterative prediction of a one-step ahead from time t+1 up to time t+K. We used the results from the previous chapters to propagate the uncertainty resulting from the noisy values which have been fed back to the model. To illustrate the performance and practicality of the proposed robust approach, we explored one simulated example as well as Mackey-Glass chaotic time series with an injected mixture of two Gaussian noises. Then, we compared the multi-step ahead prediction of the proposed method with Standard GP-based dynamical model.

Chapter 5

Conclusions

5.1 Summary of This Thesis

This thesis investigated a number of extensions to the Gaussian Process (GP), to improve the accuracy of data-driven based modelling.

In Chapter 1, we explained the frequent challenges encountered in practice for control of complex processes. These challenges motivated us to focus on Gaussian Process Regression(GPR) as a non-parametric model.

Since industrial data may be corrupted by outlying observation, Chapter 2 considers a robust GP with a mixture of two Gaussians noise model whereby one of the normal distributions with the lower variance captures the regular noises, and the other normal distribution with relatively high variance captures irregular noises. As there are two possibilities for each sample, the problem turns out to be a combinatorial problem. Thus, this robust GP model cannot be learned using maximum likelihood estimation, unlike the standard GP. An EM-based algorithm was discussed in this chapter, to construct a lower bound to maximum likelihood estimation whereby all the hyper-parameters including both GP parameters and noise parameters were learned. The performance of the proposed method was then evaluated on two synthetic datasets-Neal and Friedman datasets, and compared with other robust GP method existing in the literature. Further, a CSRT simulation example, as well as industrial datasets from SAGD process was explored to show the effectiveness of the proposed method in complex chemical processes.

In Chapter 3, a new robust model was suggested to account for the noise in inputs. Thus, we extended the proposed robust GP in Chapter 2 to the case with

noisy input. We derived an algorithm wherein the prior on the underlying process was approximated with a new Gaussian process that involves the input noise variance in its kernel function. Then, the input noise parameters along with all the other hyper-parameters were trained using the proposed algorithm based on EM approach, whereby the first and second moment of the posterior predictive distribution is computed (Gaussian approximation). To investigate the performance of the proposed method, we applied the method on a synthetic data set as well as a CSRT simulation example and compared the simulation results with standard GP.

In Chapter 4, the application of the methods from previous chapters was considered for identification of nonlinear dynamic systems. Further, we made a multiple steps ahead prediction using iterative calculations. We started with one step ahead prediction given past outputs and external inputs in addition to a NARX structure model, wherein the proposed robust GP regression was utilized to feed back the predictive distribution for each delayed output. Then, this algorithm was verified with Mackey Glass chaotic dataset and a synthetic example.

5.2 Directions for Future Work

In this thesis, a Gaussian Process was used as prior on latent function values, which is characterized by zero or constant mean function and the squared exponential kernel function. There are several avenues for further research that are listed as follows;

1. A promising future direction would be to develop our derivation for more sophisticated covariance functions or any kind of mean functions. We would also consider another method such as variational method rather than EM to derive the distribution on the latent variables instead of point estimation.

2. Another possibility for further research would be to incorporate non-stationary kernels in the proposed robust GP for the cases wherein the smoothness changes with location (so-called spatial smoothness). Another potential extension is to enhance our formulation and implementation, for multiple outputs and correlated measure outputs.

3. This thesis considered the identification of nonlinear dynamic systems using

the proposed robust GP. Another potential avenue to further research would be to use multiple robust Gaussian process models to the identification of the underlying dynamic systems mapping the large data sets.

4. Further, we can incorporate time into the model which means that for example the covariance function and mean function parameters are dependent on time. Using this extension, the time-varying dynamics system can be modelled.

5. We can also consider state-space modelling with the proposed robust Gaussian process such that the proposed robust GP can be utilized as transition models with a state-space model to get an inside of nonlinear dynamic systems for control application.

Bibliography

Wayne A Fuller. *Measurement error models*, volume 305. John Wiley & Sons, 2009.

Chi-Lun Cheng, John W Van Ness, et al. *Statistical regression with measurement error*. London: Arnold and New York: Oxford University Press, 1999.

Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

Vladimir N Vapnik. The nature of statistical learning theory. 1995.

Howard B Demuth, Mark H Beale, Orlando De Jess, and Martin T Hagan. *Neural network design*. Martin Hagan, 2014.

Danie G Krige. *A statistical approach to some mine valuation and allied problems on the Witwatersrand: By DG Krige*. PhD thesis, University of the Witwatersrand, 1951.

Anthony O'Hagan and JFC Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42, 1978.

Radford M Neal. *BAYESIAN LEARNING FOR NEURAL NETWORKS*. PhD thesis, University of Toronto, 1995.

CKI Williams. Prediction with gaussian processes: from linear regression to linear prediction and beyond. In *Learning in graphical models*, pages 599–621. MIT Press, 1999.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Radford M Neal. Bayesian learning via stochastic dynamics. In *Advances in neural information processing systems*, pages 475–482, 1993.

Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520, 1996.

Radford M Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*, 1997.

Mark N Gibbs. *Bayesian Gaussian processes for regression and classification*. PhD thesis, University of Cambridge Cambridge, England, 1998.

Tao Chen, Julian Morris, and Elaine Martin. Gaussian process regression for multivariate spectroscopic calibration. *Chemometrics and Intelligent Laboratory Systems*, 87(1):59–71, 2007.

Yi Liu, Tao Chen, and Junghui Chen. Auto-switch gaussian process regression-based probabilistic soft sensors for industrial multigrade processes with transitions. *Industrial & Engineering Chemistry Research*, 54(18):5037–5047, 2015.

Yi-Jun He, Jia-Ni Shen, Ji-Fu Shen, and Zi-Feng Ma. State of health estimation of lithium-ion batteries: A multiscale gaussian process regression modeling approach. *AIChE Journal*, 61(5):1589–1600, 2015.

Roderick Murray-smith, Carl Edward Rasmussen, and Agathe Girard. Gaussian process model based predictive control. In *In Proceedings of 4th American Control Conference*. Citeseer, 2004.

George EP Box and George C Tiao. A further look at robustness via bayes's theorem. *Biometrika*, 49(3/4):419–432, 1962.

Anthony O'Hagan. On outlier rejection phenomena in bayes inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 358–367, 1979.

Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

Mike West. Outlier models and prior distributions in bayesian linear regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 431–439, 1984.

George EP Box and George C Tiao. A bayesian approach to some outlier problems. *Biometrika*, 55(1):119–129, 1968.

Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

Malte Kuss. *Gaussian process models for robust regression, classification, and reinforcement learning*. PhD thesis, Technische Universität, 2006.

Michael E Tipping and Neil D Lawrence. Variational inference for student-t models: Robust bayesian interpolation and generalised component analysis. *Neurocomputing*, 69(1):123–141, 2005.

Jarno Vanhatalo, Pasi Jylänki, and Aki Vehtari. Gaussian process regression with student-t likelihood. In *Advances in Neural Information Processing Systems*, pages 1910–1918, 2009.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(Nov):3227–3257, 2011.

Rishik Ranjan, Biao Huang, and Alireza Fatehi. Robust gaussian process modeling using em algorithm. *Journal of Process Control*, 42:125–136, 2016.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Mark Ebden. Gaussian processes: A quick introduction. *arXiv preprint arXiv:1505.02965*, 2015.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

Carl Edward Rasmussen. *EVALUATION OF GAUSSIAN PROCESSES AND OTHER METHODS FOR NON-LINEAR REGRESSION*. PhD thesis, University of Toronto, 1996.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Yaojie Lu and Biao Huang. Robust multiple-model lpv approach to nonlinear process identification using mixture t distributions. *Journal of Process Control*, 24(9): 1472–1488, 2014.

F Guo, K Hariprasad, B Huang, and YS Ding. Robust identification for nonlinear errors-in-variables systems using the em algorithm. *Journal of Process Control*, 54: 129–137, 2017.

Sean Borman. The expectation maximization algorithm-a short tutorial. *Submitted for publication*, pages 1–9, 2004.

IB Vapnyarskii. Lagrange multipliers. *Hazewinkel, Michiel, Encyclopedia of Mathematics, Springer, ISBN*, 978:1–55, 2001.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11(Nov):3011–3015, 2010.

Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.

J Duane Morningred, Bradley E Paden, Dale E Seborg, and Duncan A Mellichamp. An adaptive nonlinear predictive controller. In *American Control Conference, 1990*, pages 1614–1619. IEEE, 1990.

Carl Edward Rasmussen. *Evaluation of Gaussian processes and other methods for non-linear regression.* University of Toronto, 1999.

Raymond J Carroll, David Ruppert, Leonard A Stefanski, and Ciprian M Crainiceanu. *Measurement error in nonlinear models: a modern perspective*. CRC press, 2006.

Dennis V Lindley. Regression lines and the linear functional relationship. *Supplement to the Journal of the Royal Statistical Society*, pages 218–244, 1947.

Gene H Golub and Charles F Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, 1980.

Petros Dellaportas and David A Stephens. Bayesian analysis of errors-in-variables regression models. *Biometrics*, pages 1085–1095, 1995.

Volker Tresp, Subutai Ahmad, and Ralph Neuneier. Training neural networks with deficient data. In *Advances in neural information processing systems*, pages 128–135, 1994.

Gao Huang, Shiji Song, Cheng Wu, and Keyou You. Robust support vector regression for uncertain input and output data. *IEEE transactions on neural networks and learning systems*, 23(11):1690–1700, 2012.

Agathe Girard and Roderick Murray-Smith. Learning a gaussian process model with uncertain inputs. *Department of Computing Science, University of Glasgow, Tech. Rep. TR-2003-144*, 2003.

Agathe Girard, Carl Edward Rasmussen, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs: Multiple-step ahead prediction. In *Technical Report TR-2002–119*. Department of Computing Science. University of Glasgow, 2002.

Agathe Girard, Carl Edward Rasmussen, Joaquin Quinonero Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. In *Advances in neural information processing systems*, pages 545–552, 2003a.

Andrew McHutchon and Carl E Rasmussen. Gaussian process training with input noise. In *Advances in Neural Information Processing Systems*, pages 1341–1349, 2011.

Cuong Tran, Vladimir Pavlovic, and Robert Kopp. Gaussian process for noisy inputs with ordering constraints. *arXiv preprint arXiv:1507.00052*, 2015.

Paul W Goldberg, Christopher KI Williams, and Christopher M Bishop. Regression with input-dependent noise: A gaussian process treatment. In *Advances in neural information processing systems*, pages 493–499, 1998.

Stephen A Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

Peter Whitle. *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksells, 1951.

George EP Box, Gwilym M Jenkins, and G Reinsel. Forecasting and control. *Time Series Analysis*, 3:75, 1970.

Lennart Ljung and Torsten Söderström. *Theory and practice of recursive identification*, volume 5. JSTOR, 1983.

L. Ljung. *System identification: theory for the user*. Prentice-Hall information and system sciences series. Prentice-Hall, 1987. ISBN 9780138816407. URL https://books.google.ca/books?id=gpVRAAAAMAAJ.

T.S. Soderstrom and P.G. Stoica. *System Identification*. Prentice Hall International Series In Systems And Control Engineering. Prentice Hall, 1989. ISBN 9780138812362. URL https://books.google.ca/books?id=X_xQAAAAMAAJ.

SA Billings and IJ Leontaritis. Identification of nonlinear systems using parameter estimation techniques. 1980.

Tor Arne Johansen, Bjarne A Foss, et al. Multiple model approaches to modelling and control. *International journal of control*, 72(7-8):575–575, 1999.

Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1):116–132, 1985.

Kumpati S Narendra and Kannan Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, 1 (1):4–27, 1990.

Alan Lapedes and Robert Farber. Nonlinear signal processing using neural networks: Prediction and system modelling. Technical report, 1987.

Jose C Principe, Alok Rathie, and Jyh-Ming Kuo. Prediction of chaotic time series with neural networks and the issue of dynamic modeling. *International Journal of Bifurcation and Chaos*, 2(04):989–996, 1992.

Roderick Murray-Smith, Tor A Johansen, and Robert Shorten. On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In *Control Conference (ECC), 1999 European*, pages 3569–3574. IEEE, 1999.

DJ Leith, WE Leithead, E Solak, and R Murray-Smith. Divide & conquer identification using gaussian process priors. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 1, pages 624–629. IEEE, 2002.

K Ažman and Jus Kocijan. Dynamical systems identification using gaussian process models with incorporated local models. *Engineering Applications of Artificial Intelligence*, 24(2):398–408, 2011.

Gregor Gregorcic and Gordon Lightbody. Local model network identification with gaussian processes. *IEEE Transactions on neural networks*, 18(5):1404–1423, 2007.

A O'HAGAN. Some bayesian numerical analysis. 1992.

Ercan Solak, Roderick Murray-Smith, William E Leithead, Douglas J Leith, and Carl E Rasmussen. Derivative observations in gaussian process models of dynamic systems. In *Advances in neural information processing systems*, pages 1057–1064, 2003.

Agathe Girard, Carl Edward Rasmussen, J Quinonero-Candela, R Murray-Smith, O Winther, and J Larsen. Multiple-step ahead prediction for non linear dynamic systemsâĂŤa gaussian process treatment with propagation of the uncertainty. *Advances in neural information processing systems*, 15:529–536, 2003b.

Andreas Damianou and Neil D Lawrence. Semi-described and semi-supervised learning with gaussian processes. *arXiv preprint arXiv:1509.01168*, 2015.

Jack Wang, Aaron Hertzmann, and David M Blei. Gaussian process dynamical models. In *Advances in neural information processing systems*, pages 1441–1448, 2006.

César Lincoln C Mattos, Zhenwen Dai, Andreas Damianou, Jeremy Forth, Guilherme A Barreto, and Neil D Lawrence. Recurrent gaussian processes. *arXiv preprint arXiv:1511.06644*, 2015a.

César Lincoln C Mattos, Zhenwen Dai, Andreas Damianou, Guilherme A Barreto, and Neil D Lawrence. Deep recurrent gaussian processes for outlier-robust system identification. *Journal of Process Control*, 2017.

Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational gaussian process state-space models. In *Advances in neural information processing systems*, pages 3680–3688, 2014.

Juš Kocijan, Agathe Girard, Blaž Banko, and Roderick Murray-Smith. Dynamic systems identification with gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

César Lincoln C Mattos, José Daniel A Santos, and Guilherme A Barreto. An empirical evaluation of robust gaussian process models for system identification. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 172–180. Springer, 2015b.

César Lincoln C Mattos, Andreas Damianou, Guilherme A Barreto, and Neil D Lawrence. Latent autoregressive gaussian processes models for robust system identification. *IFAC-PapersOnLine*, 49(7):1121–1126, 2016.

R VON MISFS. Mathematical theory of probability and statistics acad. *Press, New York*, 84, 1964.

Appendix A

Mathematical Background

A.1 Marginal and conditional probabilities of multivariate normal distribution

Assume that a D-dimensional vector $\mathbf{x}$ has a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Let $\mathbf{x}$ be partitioned such that,

$$\begin{bmatrix} \mathbf{x_A} \\ \mathbf{x_B} \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \boldsymbol{\mu_A} \\ \boldsymbol{\mu_B} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma_{AA}} & \boldsymbol{\Sigma_{AB}} \\ \boldsymbol{\Sigma_{BA}} & \boldsymbol{\Sigma_{BB}} \end{bmatrix}) \tag{A.1}$$

where $\mathbf{x}_A$ and $\mathbf{x}_B$ are two subvectors of $\mathbf{x}$. The marginal distributions for $\mathbf{x}_A$ and $\mathbf{x}_B$ is,

$$\begin{aligned} \mathbf{x_A} &\sim \mathcal{N}(\boldsymbol{\mu_A}, \boldsymbol{\Sigma_{AA}}) \\ \mathbf{x_B} &\sim \mathcal{N}(\boldsymbol{\mu_B}, \boldsymbol{\Sigma_{BB}}) \end{aligned} \tag{A.2}$$

and the conditional probability of $\mathbf{x}_A$ given $\mathbf{x}_B$ are:

$$\mathbf{x_A}|\mathbf{x_B} \sim \mathcal{N}(\boldsymbol{\mu_A} - \boldsymbol{\Sigma_{AB}}\boldsymbol{\Sigma_{BB}}^{-1}(\mathbf{x_B} - \boldsymbol{\mu_B}), \boldsymbol{\Sigma_{AA}} - \boldsymbol{\Sigma_{AB}}\boldsymbol{\Sigma_{BB}}^{-1}\boldsymbol{\Sigma_{BA}}) \tag{A.3}$$

A proof for this property was given in VON MISFS [1964].

A.2 Gaussian Integral

Let $\mathbf{y}$, $\mathbf{f}$, and $\mathbf{g}$ be vectors of size m by 1, and also $\mathbf{F}$, and $\mathbf{G}$ be covariance matrices with m by m size. Then we have:

$$\int \mathcal{N}(\mathbf{y}|\boldsymbol{f}, \boldsymbol{F}).\mathcal{N}(\mathbf{f}|\boldsymbol{g}, \boldsymbol{G})\ d\mathbf{a} = \mathcal{N}(\mathbf{y}|\boldsymbol{g}, \boldsymbol{F} + \boldsymbol{G}) \tag{A.4}$$